Lecture Notes in Computer Science 2345
Edited by G. Goos, J. Hartmanis, and J. van Leeuwen

Springer
Berlin
Heidelberg
New York
Barcelona
Hong Kong
London
Milan
Paris
Tokyo

Enrico Gregori Marco Conti
Andrew T. Campbell Guy Omidyar
Moshe Zukerman (Eds.)

NETWORKING 2002

Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communications

Second International IFIP-TC6 Networking Conference
Pisa, Italy, May 19-24, 2002
Proceedings

Springer

Volume Editors

Enrico Gregori
Marco Conti
Consiglio Nazionale delle Ricerche
Istituto di Informatica e Telematica
Via G. Moruzzi, 1, 56124 Pisa, Italy
E-mail: {enrico.gregori,marco.conti}@cnuce.cnr.it

Andrew T. Campbell
Columbia University, Department of Electrical Engineering
1312 Seeley W. Mudd Bldg., New York, NY 10027-6699, USA
E-mail: campbell@comet.columbia.edu

Guy Omidyar
National University of Singapore, Center for Wireless Communications
Singapore Science Park II, TeleTech Park
20 Science Park Road, 02-34/37, Singapore 117674
E-mail: gomidyar@cwc.nus.edu.sg

Moshe Zukerman
The University of Melbourne, EEE Department
Grattan St. Victoria 3010, Australia
E-mail: m.zukerman@ee.mu.oz.au

Cataloging-in-Publication Data applied for

Die Deutsche Bibliothek - CIP-Einheitsaufnahme

Networking 2002 : networking technologies, services, and protocols ;
performance of computer and communication networks ; mobile and wireless
communication ; proceedings / Second International IFIP TC6 Networking
Conference, Pisa, Italy, May 19 - 24, 2002. Enrico Gregori ... (ed.). -
Berlin ; Heidelberg ; New York ; Barcelona ; Hong Kong ; London ; Milan ;
Paris ; Tokyo : Springer, 2002
(Lecture notes in computer science ; Vol. 2345)
ISBN 3-540-43709-6

CR Subject Classification (1998): C.2, C.4, D.2, H.4.3, J.2, J.1, K.6, K.4

ISSN 0302-9743
ISBN 3-540-43709-6 Springer-Verlag Berlin Heidelberg New York

Springer-Verlag Berlin Heidelberg New York
a member of BertelsmannSpringer Science+Business Media GmbH

http://www.springer.de

Printed in Germany

Typesetting: Camera-ready by author, data conversion by PTP-Berlin, Stefan Sossna
Printed on acid-free paper SPIN 10869977 06/3142 5 4 3 2 1 0

Preface

This book constitutes the refereed proceedings of the Second IFIP-TC6 Networking Conference, Networking 2002. Networking 2002 was sponsored by the IFIP Working Groups 6.2, 6.3, and 6.8. For this reason the conference was structured into three tracks: i) Networking Technologies, Services, and Protocols, ii) Performance of Computer and Communication Networks, and iii) Mobile and Wireless Communications.

This year the conference received 314 submissions coming from 42 countries from all five continents Africa (4), Asia (84), America (63), Europe (158), and Oceania (5). This represents a 50% increase in submissions over the first conference, thus indicating that Networking is becoming a reference conference for worldwide researchers in the networking community.

With so many papers to choose from, the job of the Technical Program Committee, to provide a conference program of the highest technical excellence, was both challenging and time consuming. From the 314 submissions, we finally selected 82 full papers for presentation during the conference technical sessions.

To give young researchers and researchers from emerging countries the opportunity to present their work and to receive useful feedback from participants, we decided to include two poster sessions during the technical program. Thirty-one short papers were selected for presentation during the poster sessions.

The conference technical program was split into three days, and included, in addition to the 82 refereed contributions, 5 invited papers from top-level researchers in the networking community.

The technical program also included a panel session, and three invited talks from worldwide leaders – Imrich Chlamtac "Managing Optical Networks in the Optical Domain", Randy Katz "The Post-PC Era: It's All About Service", and Gerald Maguire "Personal Computing and Communication". The panel session, organized by Andrew T. Campbell (Columbia University), was entitled "Post 9-11 Networking Challenge" and is devoted to the discussion on how to cope with the vulnerabilities of communications systems revealed by the World Trade Center attack on September 11.

This conference would not have been possible without the enthusiastic and hard work of a number of colleagues. First of all, I would like to thank the three track chairs – Andrew T. Campbell, Guy Omidyar, and Moshe Zukerman – for their valuable contribution in setting up the very high quality conference program. A special thanks to the TPC members, and all the referees, for their invaluable help in reviewing the papers for Networking 2002. Finally, I would like to thank all the authors that submitted their papers to this conference for their interest and time.

March 2002 Marco Conti

Message from the General Chairs

Networking 2002 was organized by the Italian National Research Council (CNR) and Telecom Italia and was sponsored by the IFIP working groups WG 6.2 (Network and Internetwork Architectures), WG 6.3 (Performance of Communication Systems), and WG 6.8 (Wireless Communications). The program of the conference spanned on five days and included the main conference (three days), two tutorial days, and one day of thematic workshops.

The organization of such a complex event required a major effort and we wish to express our sincere appreciation to all the executive committee members for their excellent work.

We would like to express our special appreciation to the main conference technical program chair Marco Conti and to the special track chairs: Andrew T. Campbell, Moshe Zukerman, Guy Omidyar. The overall high quality of the conference technical sessions is the result of a complex evaluation process that they handled in an excellent way.

Special thanks goes to Giuseppe Anastasi and Stefano Basagni for the organization of an original and interesting tutorial program. The conference considered tutorials an important cultural event, and encouraged in several ways, the participation of young researchers in these tutorials. We decided to have a single, modest fee to provide access to all. The tutorial program included nine half-day tutorials organized in three parallel sessions.

Networking 2002 also decided to stimulate thematic events covering hot research topics in the networking field. Three thematic workshops were held: Web Engineering, Peer-to-Peer Computing, and IP over WDM. Hence our third word of thanks goes to the chairs of the thematic workshops: Fabio Panzieri, Ludmilla Cherkasova (Workshop on Web Engineering), Gianpaolo Cugola, Gian Pietro Picco (Workshop on Peer-to-Peer Computing), and Giancarlo Prati, Piero Castoldi (Workshop on IP over WDM).

We are also indebted to our supporters. First of all, CNR not only allowed Enrico Gregori and Marco Conti to dedicate considerable time to the organization of this event, but also financially supported the event through the sponsorship by the CNUCE and IIT institutes. A special thanks to Telecom Italia for joining us in the organization of this event. We are also indebted to our corporate sponsors (Cassa di Risparmio di Pisa, Compaq, Microsoft, and Softech) whose help removed much of the financial uncertainty, involved in the organization of such an event, and who also provided interesting suggestions for the program.

Our last word of gratitude goes to the Web manager Alessandro Urpi and the Web designer Patrizia Andronico. Alessandro created a very fancy and efficient system for the handling of electronic submissions. This system greatly facilitated the paper reviewing process, as well as the preparation of the proceedings. Patrizia was responsible for designing the Networking 2002 Web site that played an important role in the success of the event.

March 2002

Enrico Gregori
Ioannis Stavrakakis

Organizers

TELECOM ITALIA

Sponsoring Institutions

PISA CASSA DI RISPARMIO SPA GRUPPO BIPIELLE

COMPAQ

Microsoft®

SOFTEC

CNUCE

IIT

Organization

Conference Executive Committee

General Chair:
Enrico Gregori, National Research Council, Italy

General Vice-Chair:
Ioannis Stavrakakis, University of Athens, Greece

Technical Program Chair:
Marco Conti, National Research Council, Italy

Special Track Chair for Networking Technologies, Services, and Protocols:
Andrew T. Campbell, Columbia University, USA

Special Track Chair for Performance of Computer and Communication Networks:
Moshe Zukerman, University of Melbourne, Australia

Special Track Chair for Mobile and Wireless Communications:
Guy Omidyar, National University of Singapore

Tutorial Program Co-chairs:
Giuseppe Anastasi, University of Pisa, Italy
Stefano Basagni, Northeastern University, USA

Workshop Chairs:

Workshop 1 — *Web Engineering*
Fabio Panzieri, Università di Bologna, Italy
Ludmilla Cherkasova, Hewlett Packard Labs, USA

Workshop 2 — *Peer to Peer Computing*
Gian Pietro Picco, Politecnico di Milano, Italy
Gianpaolo Cugola, Politecnico di Milano, Italy

Workshop 3 — *IP over WDM*
Giancarlo Prati, Scuola Superiore S. Anna, Italy
Piero Castoldi, Scuola Superiore S. Anna, Italy

Invited Speaker Chair:
Fabrizio Davide, PhD Telecom Italia S.p.A., Italy

Organization Chair:
Stefano Giordano, University of Pisa, Italy

Publicity Chair:
Silvia Giordano, Federal Inst. of Technology Lausanne (EPFL), Switzerland
Laura Feeney, SICS, Sweden

Steering Committee Chair:
Harry Perros, North Carolina State University, USA

Steering Committee Members:

Augusto Casaca, IST/INESC, Portugal
S. K. Das, The University of Texas at Arlington, USA
Erol Gelenbe, University of Central Florida, USA
Harry Perros, NCSU, USA (Chair)
Guy Pujolle, University of Paris 6, France
Harry Rudin, Switzerland
Jan Slavik, TESTCOM, Czech Republic
Hideaki Takagi, University of Tsukuba, Japan
Samir Thome, ENST, France
Adam Wolisz, TU–Berlin, Germany

Electronic Submission:
Alessandro Urpi, University of Pisa, Italy

Web Designer:
Patrizia Andronico, IAT–CNR, Italy

Local organizing Committee:
Renzo Beltrame, CNUCE–CNR, Italy
Raffaele Bruno, CNUCE–CNR, Italy
Willy Lapenna, CNUCE–CNR, Italy
Gaia Maselli, CNUCE–CNR, Italy
Renata Bandelloni, CNUCE–CNR, Italy

Technical Program Committee

Special Track for Networking Technologies, Services, and Protocols

Ian Akyldiz, Georgia Institute of Technology, USA
Andrea Basso, AT&T Labs Research, USA
Edoardo Biagioni, University of Hawaii at Manoa, USA
Giuseppe Bianchi, University of Palermo, Italy
Andrea Bianco, Politecnico di Torino, Italy
Claude Castelluccia, INRIA, France
Piero Castoldi, Scuola Superiore Sant'Anna, Italy
Piergiorgio Cremonese, Netikos, Italy

Jon Crowcroft, Cambridge University, UK
Christophe Diot, Sprint, USA
Serge Fdida, Université Pierre et Marie Curie, France
Tiziana Ferrari, INFN-CNAF, Italy
Luigi Fratta, Politecnico di Milano, Italy
Maurice Gagnaire, Ecole Nationale Supérieure des Telecommunications, France
Dieter Gantenbein, IBM Research Laboratory - Zurich, Switzerland
Per Gunningberg, Uppsala University, Sweden
Salim Hariri, The University of Arizona, USA
David Hutchison, Lancaster University, UK
Bijan Jabbari, George Mason University, USA
Mohan Kumar, The University of Texas at Arlington, USA
Alfio Lombardo, University of Catania, Italy
Nicholas F. Maxemchuk, Columbia University, USA
Derek McAuley, Marconi Labs, Cambridge, UK
Refik Molva, Institut Eurécom, France
Guido H. Petit, Alcatel, Belgium
Chiara Petrioli, University “La Sapienza” Rome, Italy
Luigi Rizzo, Univeristy of Pisa, Italy
Roberto Sabella, Ericsson, Italy
Michael I. Smirnov, FHI FOKUS, Germany
Andras Valko, Ericsson, Sweden
Giorgio Ventre, Università di Napoli Federico II, Italy
Lars Wolf, University of Karlsruhe, Germany
Stefano Zatti, ESA/ESRIN, Italy

Special Track for Performance of Computer and Communication Networks

Ron Addie, University of Southern Queensland, Australia
Marco Ajmone, Marsan Politecnico di Torino, Italy
Eitan Altman, INRIA, France
Lachlan Andrew, The University of Melbourne, Australia
Andrea Baiocchi, University “La Sapienza” Rome, Italy
Chris Blondia, University of Antwerp, Belgium
Herwig Bruneel, University of Ghent, Belgium
Werner Bux, IBM Research Laboratory - Zurich, Switzerland
Mariacarla Calzarossa, University of Pavia, Italy
Olga Casals, Universitat Politecnica de Catalunya, Spain
Nelson Fonseca, State University of Campinas, Brazil
Peter Harrison, Imperial College, UK
Farouk Kamoun, Tunisia
Peter Key, Microsoft Research Ltd, Cambridge, UK
Ulf Korner, Lund University, Sweden
Demetres Kouvatsos, University of Bradford, UK

Debasis Mitra, AT&T Bell Laboratories, USA
Sandor Molnar, Budapest University of Technology and Economics, Hungary
Tim Neame, Telstra Research Laboratories, Australia
Ilkka Norros, VTT, Finland
Ramon Puigjaner, Universitat de les Illes Balears, Spain
Jim Roberts, France Telecom, France
Yutaka Takahashi, Kyoto University, Japan
Don Towsley, University of Massachusetts, USA
Phuoc Tran-Gia, University of Würzburg, Germany
Jorma Virtamo, Helsinki University of Technology, Finland
Maria C. Yuang, National Chiao Tung University, Taiwan
Bartek Wydrowski, The University of Melbourne, Australia

Special Track for Mobile and Wireless Communications:

Victor Bahl, Microsoft Research, USA
Roberto Battiti, University of Trento, Italy
Luciano Bononi, University of Bologna, Italy
Azzedine Boukerche, University of North Texas, USA
Franco Davoli, University of Genova, Italy
Khaled Elsayed, Cairo University, Egypt
Anthony Ephremides, University of Maryland, USA
Kari-Pekka Estola, Nokia Research Center, Finland
Laura M. Feeney, SICS, Sweden
Gabor Fodor, Ericsson, Sweden
Jerome Galtier, INRIA, France
Mario Gerla, University of California at Los Angeles, USA
Silvia Giordano, ICA-DSC-EPFL, Switzerland
Zygmunt Haas, Cornell University, USA
Pascal Lorenz, Université de Haute Alsace, France
Thomas Luckenback, FhG Fokus, Germany
Gerald Maguire, Royal Institute of Technology, Sweden
Stephan Olariu, Old Dominion University, USA
George Polyzos, Athens University of Economics and Business, Greece
Jiang Shengming, National University of Singapore, Singapore
Violet R. Syrotiuk, University of Texas at Dallas, USA
Ivan Stojmenovic, University of Ottawa, Canada
Terry Todd, McMaster University, Canada
Nitin Vaidya, Texas A&M University, USA
Roberto Verdone, CSITE - CNR, Italy
Jeff Wieselthier, Naval Research Laboratory, USA

Referees

Samuli Aalto
Ron Addie
Ian Akyldiz
Khalid Al-begain
Guido Albertengo
Eitan Altman

Lachlan Andrew
Csaba Antal
Panagiotis Antoniadis
Irfan Awan
Andrea Baiocchi
Dennis Baker
Mario Baldi
Mark Banfield
Chadi Barakat
Jose Barcelo
Novella Bartolini
Stefano Basagni
Andrea Basso
Roberto Battiti
Daniel Bauer
Sergio Beker
Sebastien Bertrand
Supratik Bhattacharyya
Edoardo Biagioni
Giuseppe Bianchi
Andrea Bianco
Michael Biggar
Jozsef Biro
Mats Bjorkman
Chris Blondia
Bernd Bochow
Rene Boel
Luciano Bononi
Tamas Borsos
Alessandro Bosco
Azzedine Boukerche
Onno Boxma
Rafik Braham
Hartmut Brandt
Alberto Bricca
Mauro Brunato
Raffaele Bruno
Sonja Buchegger
Laurent Bussard
Werner Bux
Mariacarla Calzarossa
Andrew Campbell
Roberto Canonico
Antonio Capobianco
Georg Carle
Olga Casals
Ramon Casellas
Claudio Casetti
Maurizio Casoni
Claude Castelluccia
Piero Castoldi
Nedo Celandroni
Llorenc Cerda
Walter Cerroni
Carla Chiasserini
Phil Chimento
Kwan-wu Chin
Chen-nee Chuah
Tibor Cinkler
Touati Corinne
Luis Costa
Piergiorgio Cremonese
Jon Crowcroft
Filippo Cugini
John Cushnie
Jeremy De Clercq
John Daigle
Olivier Dalle
Davide Dardari
Maurizio Darienzo
Bruce Davie
Franco Davoli
Stijn De Vuyst
Andrea De Vendictis
Christophe Deleuze
Francesco Delfino
Luca Dell'Uomo
Jing Deng
Ada Diaconescu
Gianluca Dini
Christophe Diot
Constantinos Dovrolis
Anca Dracinschi-sailer
Adam Dunkels
Martin Dunmore
Larry Dunn
Amre El-hoiydi
Didier Erasme
Christopher Edwards
Wolfgang Effelsberg
Viktoria Elek
Khaled Elsayed
Anthony Ephremides
Vincenzo Eramo
Alberto Escudero-Pascual
Marcello Esposito
Kari-pekka Estola
Nader Fahmy
Romano Fantacci
Laura Feeney
Meiping Feng
Tiziana Ferrari
Afonso Ferreira
Joe Finney
Paul Fitzpatrick
Gabor Fodor
Chuan Foh
Nelson Fonseca
Luigi Fratta
Laurent Frelechoux
Rod Fretwell
Hiroki Furuya
Philippe Godlewski
Dominique Grad
Maurice Gagnaire
Giulio Galante
Jerome Galtier
Dieter Gantenbein
Jorge Garcia
Rosario Garroppo
Michael Gau
Yu Ge
Mario Gerla
Vittorio Ghini
Marco Ghizzi
Paolo Giaccone
Giovanni Giambene
Chris Giblin
Silvia Giordano
Alessandra Giovanardi
Gaby Goldacker
Marcel Graf
Enrico Gregori
Fredrik Gunnarsson
Per Gunningberg

Gary Hanson
Robert Haas
Stephen Hanly
Uli Harder
Salim Hariri
Jarmo Harju
Richard Harris
Peter Harrison
Dajiang He
Jens Huenerberg
David Hutchison
Esa Hyyti
Christian Hertnagl
Gianluca Iannaccone
Lengliz Ilhem
Sandor Imre
Hazer Inalteki
Veronique Inghelbrecht
Paola Iovanna
Nando Iscra
Milosh Ivanovich
Bijan Jabbari
Laura Jackson
Yuming Jiang
Mai Jin
Josue Kuri
Ahmed Kamal
Farouk Kamoun
Holger Karl
Gunnar Karlsson
Jouni Karvo
Hiroyuki Kawano
Mitchell Ken
Csaba Keszei
Peter Key
Dr Khairy
Kalevi Kilkki
Andreas Kind
Ulf Korner
Demetres Kouvatsos
Ferenc Kubinszky
Mohan Kumar
Pirkko Kuusela
Stefan Kehler
Chia Lee
Koen Laevens
Willy Lapenna
John Larson
Pasi Lassila
Gwendal Le Grand
Jean-Yves Le Boudec
Chris Lechie
Sunj-Ju Lee
Oscar Lepe
Yuhong Li
Ben Liang
Xinhua Ling
Cati Llado
Francesco Lo Presti
Alfio Lombardo
Rui Lopes
Pascal Lorenz
Flaminia Luccio
Stefano Lucetti
Thomas Luckenback
Andrey Lyakhov
Joseph Macker
Gerald Maguire
Szabolcs Malomsoky
Dave Maltz
Roberto Mameli
Eleonora Manconi
Vincenzo Mancuso
Petteri Mannersalo
Mario Marchese
Chiani Marco
Dan Marinescu
Cristina Martello
Fabio Martignon
Piergiulio Maryni
Laurent Mathy
Nicholas Maxemchuk
Derek Mcauley
Octavio Medina
John Mellor
Michael Menth
Michela Meo
Bernard Metzler
Pietro Michiardi
Gyorgy Miklos
Enrico Milani
Jens Milbrandt
Debasis Mitra
Gergely Molnar
Sandor Molnar
Refik Molva
Tim Moors
Giacomo Morabito
Sayandev Mukherjee
Rami Mukhtar
Maurizio Munafo
Pars Mutaf
Gaurav Navlakha
Tim Neame
Giovanni Neglia
Marcel Neuts
Gam Nguyen
Saverio Niccolini
Ilkka Norros
Antonio Nucci
Eeva Nyberg
Stephan Olariu
Ertan Ozturk
Dimitri Papadimitriou
Fabrice Poppe
Panagiotis Papadimitratos
Dina Papagianaki
Davide Parisi
Laurence Park
Gianni Pasolini
Andrea Passarella
Tao Peng
Antonio Pescapë
Fabien Petitcolas
Alexandru Petrescu
Chiara Petrioli
Dimitrios Pezaros
Tom Pfeifer
George Polyzos
Francesco Potorti
Fabio Pugini
Ramon Puigjaner
Guy Pujolle
Rudesindo Queija
Nicholas Race

Andras Racz
Carla Raffaelli
Jianqiang Rao
Christoph Reichert
Franklin Reynolds
Jose Rezende
Fabio Ricciato
Ad Ridder
Herve Rivano
Romeo Rizzi
Luigi Rizzo
Jim Roberts
Vincent Roca
Marco Roccetti
Hermann Rohling
Simon Romano
Miklos Ronai
Sean Rooney
Yves Roudier
George Rouskas
Alain Roy
Romit Roychoudhury
Giuseppe Ruggeri
Jussi Ruutu
Winston Seah
Mike Sexton
Roberto Sabella
Stefano Salsano
Elio Salvadori
Prince Samar
Volker Sander
Takashi Sasaki
Durga Satapathy
Paolo Scotton
Nabil Seddigh
Ahmed Sehrouchni
Faisal Shad
N. Shankaranarayanan
Charles Shen
Jiang Shengming
Kasahara Shoji
Steven Simpson
Dorgham Sisalem
Tara Small
Michael Smirnov
Paul Smith
Sergios Soursos
Kathleen Spaey
Dirk Staehle
George Stamoulis2
Burkhard Stiller
Ivan Stojmenovic
Moon Sue
Violet Syrotiuk
Csanad Szabo
Istvan Szabo
Robert Szabo
Wayne Szeto
Marco Tacca
Nina Taft
Yutaka Takahashi
Christina Tavoularis
Ben Teitelbaum
David Thornley
Neame Tim
Ilenia Tinnirello
Carsten Tittel
Terry Todd
Petia Todorova
Samir Tohme
Samir Tohme2
Don Towsley
Velio Tralli
Phuoc Tran-gia
Linh Truong
Jaidi Tuah
Zoltan Turanyi
Kurt Tutschku
Alessandro Urpi
Masafumi Usuda
Peter Vetter
Mickey Vucic
Francesco Vacirca
Nitin Vaidya
Luca Valcarenghi
Benny Van Houdt
Vasos Vassiliou
Giorgio Ventre
Roberto Verdone
Andras Veres
Rolland Vida
Attila Vidacs
Jorma Virtamo
Thiemo Voigt
Hai Le Vu
Krzysztof Wajda
Joris Walraevens
Eric Wang
Andreas Wespi
Jeff Wieselthier
Lars Wolf
Mike Woodward
Bartek Wydrowski
Yang Xue
George Xylomenos
Miki Yamamoto
Jackson Yin
Maria Yuang
Gergely Zaruba
Stefano Zatti
Artur Ziviani
Moshe Zukerman

Table of Contents

Multicasting I

Differentiated Services I

Network Performance I

Self-Organizing Networks: Services and Protocols

Call Admission Control

Voice/Video Performance Modeling

Web Access

Optical Networks

Network and Traffic Modeling

Ad Hoc Networks

Resource Allocation I

LAN and PAN

Performance of Wireless Networks

Multimedia

Transmission Control Protocol (TCP)

Future Wireless Networks I

Internet Protocol (IP)

Queueing Models

Satellite Networks

Resource Allocation II

Performance of Optical Networks

Future Wireless Networks II

Multiprotocol Label Switching (MPLS)

Networks Performance II

Multicasting II

Posters Session

Channel Islands in a Reflective Ocean: Large Scale Event Distribution in Heterogeneous Networks

Jon Crowcroft

University of Cambridge
Computer Laboratory
William Gates Building
J J Thomson Avenue
Cambridge
CB3 0FD
Jon.Crowcroft@cl.cam.ac.uk

Abstract. This is a discussion paper about the possible future use of network and transport level multicast services to support extremely large scale event distribution.

To date, event notification services[40] have been limited in their scope due to limitations of the infrastructure At the same time, Internet network and transport layer multicast services have seen limited deployment due to lack of user demand (with the exception more recently of streaming services, e.g. on Sprint's US core network, and in the Internet II). Recent research in active and reflective middleware suggests a way to resolve these two problems at one go.

Event-driven and messaging infrastructures are emerging as the most flexible and feasible solution for enabling rapid and dynamic integration of legacy and monolithic software applications into distributed systems. Event infrastructures also support deployment and evolution of traditionally difficult-to-build active systems such as large-scale collaborative environments and mobility aware architectures.

Event notification is concerned with propagation of state changes in objects in the form of events. A crucial aspect of events is that they occur asynchronously. Event consumers have no control over when events are triggered. On the other hand, event suppliers do not generally know what entities might be interested in the events they provide. These two aspects clearly define event notification as a model of asynchronous and de-coupled communication, where entities communicate in order to exchange information, but do not directly control each other.

The IETF is just finishing specifying a family of reliable multicast transport protocols, for most of which there are pilot implementations. Key amongst these for the purposes of this research is the exposure to end systems of router filter functionality in a programmable way, known as *Generic Router Assist*. This is an inherent part of the Pragmatic General Multicast service, implemented by Reuters, Tibco and Cisco in their products, although it has not been widely known or used outside of the *TIBNET* products until very recently.

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 1–9, 2002.

The goal of this paper is to describe a reflective middleware system that integrates the network, transport and distributed middleware services into a seamless whole.

The outcome of this research will be to integrate this 'low-level' technology into an event middleware system, as a toolkit as well as evaluation of this approach for massive scale event notification, suitable for telemetry, novel mobile network services, and other as yet unforeseen applications.

1 Background and Introduction

The last decade has seen the great leaps in the maturity of distributed systems middleware, and in one particular area in support of a wide variety of novel applications, event notification systems. Current work on event notification middleware[39][40][41], has concentrated on providing the infrastructure necessary to enable content-based addressing of event notifications. These solutions promote a publish-subscribe-match model by which event sources publish the metadata of the events they generate, event consumers register for their events of interest passing event filter specifications, and the underlying event notification middleware undertakes the event filtering and routing process. Solutions differ usually on whether they undertake the filtering process at the source or at an intermediary mediator or channel in which the event filtering takes place. The trade-off lies on whether to increase the computational load of sources and decrease the network bandwidth consumption, or minimise the extra computational load on the sources and outsource the event filtering and routing task to a mediator component (hopefully located close to the source). All of these solutions do not leverage on the potential benefits that event multicasting to consumers requiring the same type of events, and applying very similar filters could bring. They usually require an individual unicast communication per event transmitted.

At the same time, the underlying network has become very widespread. New services such as IP multicast are finally seeing widespread deployment, especially in core networks and in intranets.

The combination of these two technologies, event services and multicast, originates historically with Tibco[20], a subsidiary of Reuters. However, their approach is somewhat limited as it takes a strict layered approach.

At the highest level, there is a publish/subscribe system, which in *TIBNET* uses *Subject Based Addressing* and *Content Based Addressing.* Receivers subscribe to subjects. The Subject is used to hash to a multicast group. Receivers subscribe to a subject but can express interest by declaring filters on content. The *TIBNET* system is then hybrid. In the wide area, IP multicast is used to distribute all content on a given subject topic to a set of site proxy servers. The site proxy servers then act on behalf of subscribers at a site and filter appropriate content out of each subject stream and deliver the remains to each subscriber.

Between the notification layer and the IP layer there is a transport layer, called Pragmatic General Multicast. To provide semi-reliable, in-order delivery, the subject messages are mapped onto PGM[10] messages, which are then multicast in IP packets. PGM provides a novel retransmission facility which takes advantage of router level “nack aggregation” (which itself prevents message implosion towards the event source), to provide filtering[15][16] of retransmissions so that only receivers missing a given message sequence number, receive it. The PGM protocol is essentially a light weight signaling protocol which allows receivers to install and remove filters on parts of the message stream. The mechanism is implemented in Cisco and other routers that run IP multicast. The end system part of the protocol is available in all common operating systems.

Almost all other event notification systems have taken the view that IP multicast was rarely deployed[1], and that the overheads in the group management protocols were too high for the rate of change of interest/subscription typical in many applications usage patterns.

Instead, they have typically taken an alternative approach of building a server level overlay for event message distribution. Recent years have seen many such overlay attempts[22] [23] [24] [25] [26] [27] [28] [29] [30]. These have met with varying degrees of success. One of the main problems of application layer service location and routing is that the placement of servers does not often ,match the underlying true topology of the physical network, and is therefore unable to gain accurate matching between a distribution tree and the actual link throughput or latencies. Nor is the system able to estimate accurately the actual available capacity or delay. Even massive scale deployments such as Akamai[31], for example, do not do very well.

Secondly, the delays through application level systems are massively higher than those through routers and switches (which are after all designed for packet forwarding, rather than server or client computation or storage resource sharing). The message is that overlays and measurement are both hard to optimise, and inefficient.

We see a number of advantages in continuing forward from where Tibco left off in integrating efficient network delivery through multicast, with an event notification service including:

Scale. We obviate the need to deploy special proxy servers to aid the distribution.

Throughput. We will be able therefore to distribute many more events per second.

Latency. Event distribution latency will approximate the packet level distribution delay , and will avoid the problems of high latency and jitter incurred when forwarding through application level processes on intermediaries.

There are two ideas we will draw from in moving forward. Firstly we will exploit advances in the network support for multicast, such as Generic Router Assist service in the PGM router element in IP multicast. Secondly, we will carry

[1] Ironically, this view was fuelled partly by a report by Sprint[21], when in fact the entire Sprint IP service supports multicast and they have at least 3500 commercial customers streaming content.

out research in ways to distribute an open interface to the multicast tree computation that IP routers implement. The way we propose doing this is through reflection.

Reflection is becoming commonplace in middleware[32] [33] [34], but has not been applied between application level systems and network level entities to our knowledge. The intent here is to offer a common API to both the multicast service, and the filtering service, so that the event notification module implementor need not be aware which layer is implementing a function.

We would envisage an extremely simple API, viz:

```
Create(Subject)
Subscribe/Join(Subject)
Publish/Send(Subject, Content)
Receive(Subject, Content Filter Expression)
```

The router level will create both a real distribution tree for subjects, and a sub-tree for each filter or merged filter set. This will be done with regard to the location (and density) of receivers. It is possible that we can use an multicast tunnel or multicast address translation service such as the one described in[11], to provide further levels of aggregation within the network. This will require the routers to perform approximate tree matching algorithms.

1.1 Solution, and Proposed Experiment

The approach we will take in the work is one of “build and learn”. We will build a piece of reflective middleware that is a shim between an existing event notification service and the reflective routing and filter service.

This will involve extending the PGM *signaling* protocol that installs and activates (via IP router alerts) the filters.

We will also investigate efficient hashes for subject to group and content to sequence number mapping.

Subsequently, we aim to evaluate our approach by applying it to a large-scale event driven (sentient) application, such as novel context-aware applications for the emerging UMTS mobile telephony standard[37] or large-scale location tracking applications[38]. For example, there is the possibility of developing a location tracking (people, vehicles and baggage) for large new airport terminals.

2 Overlays and Reflection

As we can see, what we are designing is effectvely a two-tier system, which entails multicast trees, and within these, filters. To these, we believe we have to add a third layer, which is illustrated in figure 1.

The purpose of the overlay is to accomodate a varieity of qualitative heterogeneity, where the lower two layers of multicast and filtering target the area of quantitative performance differences.

Firstly, initial event systems are built without any notion of a multicast filter-capable transport. Thus we must haev an overlay of event distribute servers. These can, where the lower services are available, be programmed to take advantage of it, *amongt themselves*, thus providing a seemless mechanism to deploy the new service transparently to publisher and subscriber systems. However, we also believe that there are *inherent* structural reasons why such an applicaiton layer overly is needed. These include:

Policies. Different regions of the network will have different policies about which events may be published and which not.

Security. There may be firewall or other security mechanisms which impede the distribution via lower level protocols.

Evolution. We would like to accomodate evolution (in the same way that inter-domain routing protocols such as BGP allow intra-domain routing to eveolve).

Interworking. We would like to accomodate multiple event distribution middleware.

Others. There are other such "impedence mismatches" which we may encounter as the system scales up.

A novel aspect of our approach is that the overlay system does not, itself, construct a distribution tree. Isntead, a set of *virtual* members are addd to the lower level distribution system whcih then uses its normal multicast routign algorithms to cnstruct a distributio ntree amongstr a set of event notificaiton servers seperated in islands of multicast capable networks. These servers then use an open interface to quret the routers as to the computed tree, and then use this as their own distribtion - in this way the overlay can take advanatage of detaield metric information that the router layer has access to (such as delay, throughput and current load on links) instead of measuring a poor shadow of that data which would lead to, an inaccurate and out of date parameters with which to build the overlay. In some senses, what we are doign here is like multicast traffic engineering!

We believe that our system provides a number of engineering performance enhancements over previous event notificaiton architectures. Future work will evaluate these, which include:

1. System performance - improvement in scalability, including reduction in join/leave publish/subscribe latency, increase in event throughput, etc.
2. Network impact - impact on router load by filter cost group join, leave and multicast packet forwarding.
3. Expressiveness and seamlessness of API - try it with variety of event notification systems! export via public CVS and see what open source community do?

3 Discussion

For now, its an idea, but we can envisage a world in which perveasive computing devices generate 10,000,000,000 events per second. We can foresee a time when there are thousands of millions of event subscribers all over the planet, with publishers having popularities as low as no or only a single subscriber, or as high as the entire world.

One of the goals of this system is to explore the way that the multicast treees evolve and the filtering system evolves. Another goal is to see how multicast routign can be "laid open" as a service to be used to build distribution trees for other layers. Fianlyl, we belieev that the three levels we have may not be enough, and that as the system grows larger still, other services may emerge.

It is frequently the case that in the long term, business migrates into the infrastructure. (c.f. voice, IP, etc). We expect many overlay services to do this. We believe that this process by will accelerate due to use of state of the art network, middleware and software engineering approaches. However, this process will not stop - there is an endless stream of new services being introduced "at the top", and makign their way down to the bottom, to emerge as part of the critical information infrastructure.

The architecture is illustrated in figure 1. In this we can see that a publisher creates a sequence of events, which carry attributes with given values. A consumer subscribes to a publisher, and may express content based filters to the publisher. In our system, these filter expressions can be distributed up-stream from the consumer towards the publisher. As they pass through Application-

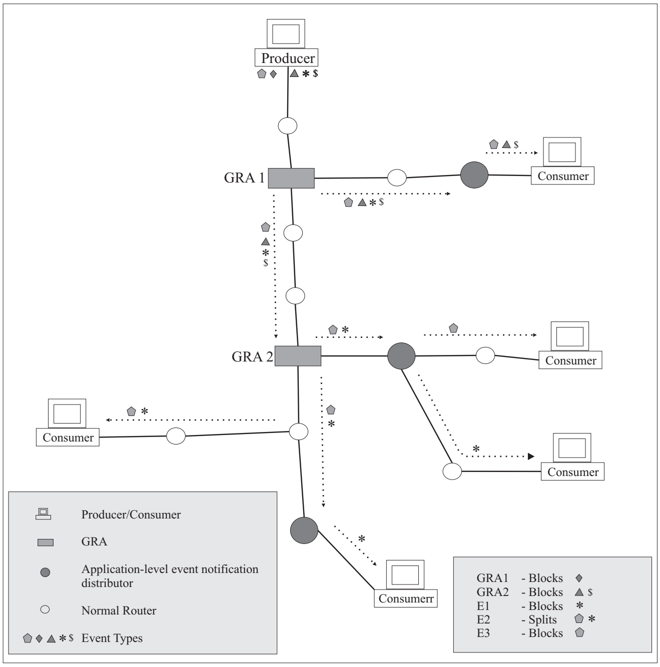


Fig. 1. Channel Islands System Architecture

level event notification distributers, they can be evaluated and compared, and possibly combined with other subscription filters. Notifications of interest are passed up stream all the way to the publisher, or to the application-level cevent notification distributer nearest the publiser, which can then compute a set of fixed tags for data; it can also, by consulting with the IP and GRA routers, through the reflective multicast routing service, compute a set of IP multicast groups over which to distribute the data, which will create the most efficient trade-off between source and network load, and receiver load, as well as tag and filter evaluation, as the events are carried downstream from the publiser, over the IP multicast, GRA, and application-level event notification nodes. Devising and evaluating the detailed performance of the algorithms to carry out these tasks out form the core of the requirements for future work.

Acknowledgements. The author gratefully acknowledges discussions with his colleagues, particularly Jean Bacon and George Coulouris.

References

[1] A. Mankin, A. Romanow, S. Bradner and V. Paxson, “IETF Criteria for Evaluating Reliable Multicast Transport and Application Protocols” RFC2357, June 1998.

[2] Reliable Multicast Research Group `http://www.east.isi.edu/RMRG/`

[3] S.Floyd, V.Jacobson, C.Liu, S.McCanne, L. Zhang, “A Reliable Multicast Framework for Light-weight Sessions and Application Level Framing, Scalable Reliable Multicast (SRM)”, ACM SIGCOMM’95.

[4] M.Handley and J.Crowcroft, “Network Text Editor (NTE): A scalable shared text editor for the Mbone”, ACM SIGCOMM’97, Cannes, France, September 1997.

[5] “TCP-like Congestion Control for Layered Multicast Data Transfer”, L.Vicisano, L.Rizzo, J.Crowcroft, INFOCOM’98.

[6] “IEEE Standard for Distributed Interactive Simulation - Application Protocols” IEEE std 1278.1-1995, IEEE Computer Society

[7] “IEEE Standard for Distributed Interactive Simulation - Communications Services and Profiles”, IEEE std 1278.2-1995, IEEE Computer Society

[8] Mark Handley et al, Building Blocks for Reliable Multicast Transport Protocols, Work in progress, RMT Working Group, IETF.

[9] “Rate Adjustment Protocol” Handley, M. et al Proc Infocom 1999, NY

[10] Pragmatic Generalised Multicast Tony Speakman, et al, Work in Progress, `http://search.ietf.org/internet-drafts/draft-speakman-pgm-spec-07.txt`

[11] “Multicast Address Translation” Work in Progress, `http://www.ietf.org/internet-drafts/draft-crowcroft-mat-00.txt`

[12] “Self Organising Transcoders”, Kouvelas, I. et al Proc NOSSDAV 1998, Cambridge England

[13] “Router Mechanisms to Support End-to-End Congestion Control”, S.Floyd, K.Fall, Technical report, `ftp://ftp.ee.lbl.gov/papers/collapse.ps`.

[14] “RMTP: A Reliable Multicast Transport Protocol”, J.C. Lin, S.Paul, IEEE INFOCOM ’96, March 1996, pp.1414-1424.
Available as `ftp://gwen.cs.purdue.edu/pub/lin/rmtp.ps.Z`

[15] "Generic Router Assist Building Block", B. Cain, T. Speakman, D. Towsley, Internet Drafts, Work in progress. `http://search.ietf.org/internet-drafts/draft-ietf-rmt-gra-fspec-00.txt` and `http://search.ietf.org/internet-drafts/draft-ietf-rmt-gra-arch-02.txt`

[16] GMTS "Generic Multicast Transport Services" B. Cain, D. Towsley, in Proc. Networking 2000, Paris, France May 2000. `http://www.east.isi.edu/RMRG/cain-towsley3/`

[17] "Incremental Depoyment of a Router-assisted Relaible Multicast Scheme" C. Papadopoulos, E. Laliotis Proc of NGC 2000 WOrkshop.

[18] "COBEA: A CORBA-Based Event Architecture" C. Ma and J. Bacon Proc of 4th Usenix Conference on Object Oriented Technologies and Systems, 1998

[19] "Building Event Services on Standard Middleware" Jean Bacon, Alexis Hombrecher, Chaoying Ma, Ken Moody, Peter Pietzuch Work in Progress.

[20] TIBCO `http://www.tibco.com`

[21] "Deployment Issues for the IP Multicast Service and Architecture", C. Diot, B. N. Levine, B. Lyles, H. Kassem, D. Balensiefen. IEEE Network magazine special issue on Multicasting. January/February 2000.

[22] "A Case For End System Multicast", Y. Chu, S. Rao, H. Zhang, Proceedings of ACM SIGMETRICS , Santa Clara,CA, June 2000, pp 1-12.

[23] "Enabling Conferencing Applications on the Internet Using an Overlay Multicast Architecture" Y. Chu, S. Rao, S. Seshan, H. Zhang, Proc. ACM Sigcomm 2001, `http://www.acm.org/sigs/sigcomm/sigcomm2001/p5-chu.pdf`

[24] "Overcast: Reliable Multicasting with an Overlay Network", J. Jannotti, D. K. Gifford, K. L. Johnson, M. F. Kaashoek, and J. W. O'Toole, Jr., Proceedings of OSDI'00. `http://gaia.cs.umass.edu/cs791n/Jannotti00.pdf`

[25] "Tapestry: a fault tolerant wide area network infrastructure", B. Zhou, D. A. Joseph, J. Kubiatowicz, Sigcomm 2001 poster and UC Berkeley Tech. Report UCB/CSD-01-1141. `http://www.cs.berkeley.edu/ ravenben/publications/CSD-01-1141.pdf`

[26] "Chord: A Scalable Peer-To-Peer Lookup Service for Internet Applications" I. Stoica, R. Morris, D. Karger, F. Kaashoek, H. Balakrishnan, ACM Sigcomm2001, `http://www.acm.org/sigcomm/sigcomm2001/p12.html`

[27] S. Ratnasamy, P. Francis, M. Handley, R. Karp, S. Shenker, "A Scalable Content-Addressable Network" ACM Sigcomm 2001, `http://www.acm.org/sigcomm/sigcomm2001/p13.html`

[28] "Application-Level Anycasting: a Server Selection Architecture and Use in a Replicated Web Service" E. Zegura, M. Ammar, Z. Fei, and S. Bhattacharjee. IEEE/ACM Transactions on Networking, Aug. 2000. `ftp://ftp.cs.umd.edu/pub/bobby/publications/anycast-ToN-2000.ps.gz`

[29] "Evaluation of a Novel Two-Step Server Selection", K. M. Hanna, N. Natarajan, and B.N. Levine, Metric To Appear in IEEE ICNP 2001. November 2001. `http://www.cs.umass.edu/ hanna/papers/icnp01.ps`

[30] "Finding Close Friends on the Internet" Christopher Kommareddy, Narendar Shankar, Bobby Bhattacharjee, To appear in ICNP 2001.

[31] "An Investigation of Geographic Mapping Techniques for Internet Hosts" Venkata N. Padmanabhan, Lakshminarayanan Subramanian, Proc of ACM SIGCOMM 2001, San Dieogo, 2001. `http://www.acm.org/sigcomm/sigcomm2001/p14.html`

[32] “Integrating Meta-Information Management and Reflection in Middleware”, Fabio Costa and Gordon Blair 2nd International Symposium on Distributed Objects & Applications pp. 133-143, Antwerp, Belgium, Sept. 21-23, 2000. Internal report number MPG-00-20

[33] “The Role of Open Implementation and Reflection in Supporting Mobile Applications ” Gordon Blair Proceedings of the IEEE Workshop on Mobility in Databases and Distributed Systems (MDDS’98), Vienna, August 1998. Internal report number MPG-98-35.

[34] “Open Implementation and Flexibility in CSCW Toolkits”, Paul Dourish, PhD Thesis, 1996, Supervisor, Jon Crowcroft Available from ftp://cs.ucl.ac.uk/darpa/dourish-thesis.ps.gz

[35] “A Language-Based Approach to Programmable Networks”, Ian Wakeman, Alan Jeffrey and Tim Owen, IEEE Conference on Open Architectures and network Programming, March 2000, Tel-Aviv, Israel.

[36] What is Reflective Middleware? Geoff Coulson http://computer.org/dsonline/middleware/RMarticle1.htm

[37] “UMTS Networks: Architecture, Mobility and Services”, Wiley & Sons. 2001; ISBN: 047148654X, Heikki Kaaranen (Editor), Siamäk Naghian, Lauri Laitinen, Ari Ahtiainen, Valtteri Niemi

[38] The Graticule System http://www.graticule.com/products/MapGPS.html

[39] “A Survey of Event System”, A. Rifkin and R. Khare. http://www.cs.caltech.edu/ adam/isen/event-systems.html

[40] “Notification Service Specification”, Object Management Group, June 2000, ftp://ftp.omg.org/pub/docs/formal/00-06-20.pdf

[41] “Design and evaluation of a wide-area event notification service”, Carzaniga A., Rosenblum D. S. and Wolf A. L. ACM Transactions on Computer Systems, Volume 19, no. 3, pp. 332-383, 2001

A Reliable Multicast Protocol with Delay Guarantees

Nicholas F. Maxemchuk

Columbia University, Dept. of Elec. Eng., New York, N.Y.
nick@ee.columbia.edu

Abstract. The reliable multicast protocol guarantees that all receivers place the source messages in the same order. We have changed this protocol from an event driven protocol to a timed protocol in order to also guarantee that all of the receivers have the message by a dead line. In this work we present two modifications to the timed protocol that provide shorter deadlines. In the examples that we consider the tighter deadlines approach the nominal network delay.

1 Introduction

The Internet uses very simple protocols in the core of the network and relegates many functions to the end user. This strategy makes it possible to introduce new services by changing the programs at the users that require the services, rather than changing the entire network.

The Internet provides best effort delivery. It does not guarantee the message delay or that the message will be delivered at all. In order for the end user to guarantee that messages are delivered within a certain interval, the end user must have a concept of time and take action within the interval. In conventional ARQ protocols the source users a timer to periodically retransmit a message until it receives a response from the receiver. Alternatively, if the source transmits at known times, a receiver that has a clock and knows the source schedule can take action when messages aren't received. Periodic updates have been used in point to point transport protocols [1]. Recently, time has been added to the reliable broadcast protocol [2], RBP, to guarantee that all of the receivers have a message in a specified interval [3]. In the modified protocol messages are acknowledged according to a schedule and the receivers use absolute time to recover missing acknowledgements and source messages. Receivers that receive the acknowledgements and source messages do not have to send any further messages.

RBP was invented in 1984. This protocol used as few as one control message for each broadcast message, independent of the number of receivers, to guarantee that all of the receivers correctly received a broadcast message. In addition to guaranteeing that all of the receivers correctly receive every broadcast message, it guarantees that every receiver places the broadcast messages in the same sequence.

RBP was originally used to build a distributed database on an Ethernet[4]. In the early 90's, this protocol was adapted to operate on a multicast network over the Internet and was renamed the reliable multicast protocol[5], RMP.

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 10–27, 2002.

RMP is event driven. The receivers do not take any action until a message is received. The protocol guarantees that all of the receivers "eventually" receive a message, rather than guaranteeing when they receive the message. If there are N receivers, the protocols guarantees that all of the receivers have a message after $N - 1$ additional messages have been acknowledged. RMP is described in section 2.

In 1999 RMP was applied to an international, distributed stock market[3]. By adding a knowledge of absolute time to the protocol, and making the protocol time driven, rather than event driven, the earlier characteristics of RMP are maintained while also guaranteeing that every receiver receives every broadcast message within a specified time. The timed version of RMP, T-RMP, is described in section 3.

T-RMP periodically sends a control message that simultaneously acknowledges all of the unacknowledged source messages. All of the receivers know when a control message is scheduled to be transmitted and begin the recovery process soon after the scheduled transmission time, rather than waiting for a message. Once the control message is received, the receivers request any missing source messages that it acknowledged. When the period between control messages equals the average interarrival time of source messages one source message is acknowledged by each control message, on the average, and the message efficiency of T-RMP and RMP is the same. When the period between control messages is greater than the average interarrival time of source messages, more than one source message is acknowledged by each control message, and the efficiency of T-RMP is higher than RMP. However, as the period between control messages decreases, the message efficiency of T-RMP also decreases.

The version of T-RMP that is used in the stock market application is relatively easy to understand because the period between control messages is large enough for all of the receivers that have missed the control message or any of the source messages that it acknowledged to recover those messages before the next control message is transmitted. We can guarantee that the control message period is large enough for a receiver to recover a missing message because the ARQ protocol is not open ended. After a fixed number of attempts, the requesting site assumes that the site with the message has failed and enters a reformation process. Therefore, at the end of each control message period either all of the operable receivers have all of the acknowledged messages, or the system has entered a reformation process to identify failed sites.

The reformation process is a lengthy process. In order to prevent the protocol from performing a reformation when the network experiences slightly longer than normal delays, the message recovery time is much greater than the average message delay in the Internet.

The control message period is the guaranteed delivery delay for an acknowledged message. The delay guarantee that is provided by the original version of T-RMP is adequate for the stock market application, but reducing the delay will make the protocol applicable to a larger class of applications, such as remote classrooms where students ask questions.

One way to reduce the control message interval is to reduce the time between retry attempt to recover a lost message. As we make the retry intervals smaller we can take advantage of the small delays that usually occur in the network. However, the smaller retry intervals result in more frequent retries when a message has not been lost but is only delayed by the network. As the retry interval goes to zero, the time to recover a message can track the distribution of delays in the network, but the number of retries, and hence the number of overhead messages, becomes large. This effect occurs for all ARQ protocols that are used on the Internet, or any other network with variable delays. The effect is not unique to T-RMP and is not investigated in this paper.

In sections 4 and 5 we consider two ways to reduce the guaranteed delivery time that are unique to T-RMP. The original version of T-RMP uses separate retry counters to recover the control message and the source messages that it acknowledged. In section 4 we combine the counts and show that we can significantly reduce the control message period without increasing the probability of erroneously entering the reformation process. In the original version of T-RMP the control message interval, the time until a message is recovered by all of the receivers, and the time to enter the reformation process, are all the same. In section 5 we consider using different time intervals for each of these events. The operation of the protocol is more complicated. We show that the protocol operates as a D/G/1 queue and show, by an approximate analysis of the queue, that using different intervals fro the three events can significantly reduce the delay guarantees.

2 The Reliable Multicast Protocol

RMP has three characteristics that distinguishes it from earlier protocols:

1. Every receiver places the messages from the sources in the same sequence.
2. Every receiver eventually knows that every other receiver has the data.
3. When there aren't any losses, there is only one control message per source message, independent of the number of receivers. (In reference 6 there is an analysis of the number of messages that are transmitted when there are losses.)

The RMP protocol has two parts. The first part operates on multicast messages during normal operation. It guarantees delivery and ordering of the messages from the sources. The second part is a reformation protocol that reorganizes the broadcast group and guarantees the consistency of message sequences at the receivers after failures and recoveries. The complete protocol is described in reference 2. In this presentation we are concerned with the first part of the protocol.

There are n sources and m receivers that participate in the protocol, as shown in figure 1. The sources and receivers may be the same or different. A single receiver, called the token site, acknowledges a source message and assigns the message a sequence number. All of the receivers place the messages in the order indicated by the sequence number.

We guarantee that every receiver has all of the messages by sequentially passing the token to each receiver. A receiver does not accept the token until it acquires all of the preceding acknowledgments and the messages that they acknowledged. Therefore, when the receiver with the token sends an explicit acknowledgment for a source message, it implicitly acknowledges that it has received all of the source messages that have been acknowledged prior to this message.

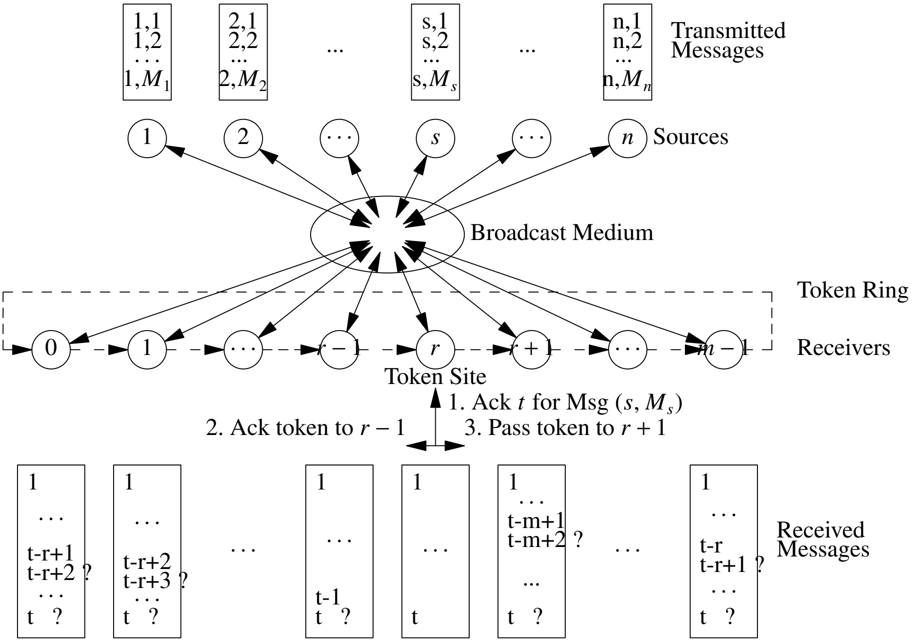


Fig. 1. The Reliable Broadcast Protocol

The sources use a positive acknowledgment protocol. A message from source s contains the label (s, M_s) to signify that it is the M_s^{th} message from source s. Source s transmits message M_s at regular intervals until it receives an acknowledgment or decides that the token site is not operating. If a source decides that the token site is not operating it initiates a reformation.

The receivers take turns acknowledging messages from sources by passing a token. A single control message, acknowledgment t from receiver r, serves three separate functions:

1. it acknowledges (s, M_s) and assigns it sequence number t,
2. it is an acknowledgment to receiver $(r-1) \bmod m$ that the token was successfully transferred to r, and,
3. it transfers the token to receiver $(r+1)$ mod m.

The token transfer uses a positive acknowledgment protocol. Token site r periodically sends acknowledgment t until it receives acknowledgment $t+1$ or greater or it receives a separate token acknowledgment. If the acknowledgment isn't received in a specified number of attempts, receiver r decides that receiver $r+1$ is inoperable and initiates a reformation.

When r sends acknowledgment t it stops acknowledging source messages, even though receiver $(r+1)$ mod m may not have received, or may not be able to accept the token. This guarantees that at most one receiver can acknowledge source messages.

When a receiver accepts the token it also assumes responsibility for servicing retransmission requests. Receiver $(r+1)$ mod m does not accept the token until it has all of the acknowledgments and source messages that were acknowledged up to and including t. Receiver r does not stop servicing retransmission requests until it receives the acknowledgment for passing the token. This guarantees that there is always at least one site, that has all of the source and control messages, that is responding to retransmission requests.

Receivers place the messages in the sequence assigned by the acknowledgments. Each receiver, r, tracks t_r, the next acknowledgment that it expects. If an acknowledgment number greater than t_r is received, acknowledgment t_r is missing. If acknowledgment t_r is received and the source message that is acknowledged is not in the receiver's queue of unacknowledged messages, then the source message is missing. The receivers use a negative acknowledgment strategy. No control messages are sent unless a missing message is detected. When a receiver detects a missing message it recovers the message using a positive acknowledgment protocol. The receiver periodically requests the message until it receives the message or decides that the retransmit server is inoperable and initiates a reformation.

As the token is passed, the token site can infer information about the other receivers. When receiver r transmits acknowledgment t, receiver r and any receiver that receives the acknowledgment knows that

— receiver r has all of the acknowledged messages up to and including the t^{th} message,

— receiver $(r-1) \bmod m$ has all of the acknowledged messages up to and including the $(t-1)^{th}$ message,

— $\cdots$, and

— receiver $(r-m+1) \bmod m$ has all of the acknowledged messages up to and including the message acknowledged by $t-m+1$.

Since $(r-m) \bmod m = r$, receiver r knows that all of the receivers have all of the source messages up to and including the message acknowledged by $t-m+1$. By a similar argument all of the receivers know that all of the other receivers have all of the messages up to and including the $(t-m+2)^{th}$ message.

Figure 2 is an extended finite state machine, E-FSM, representation of the actions that a receiver takes when an acknowledgment is processed. The states indicate tests that are performed or situations where the receiver waits for an external stimuli, such as a message or a time out. The transitions between states are labeled with the event that caused the transition, followed by a "*"'ed list of actions that occur during the transition.

3 The Timed Reliable Multicast Protocol

T-RMP uses the same token passing mechanisms and retransmission strategies as RMP, as shown in figure 1. The difference is that T-RMP is time driven rather than event driven. Acknowledgments are transmitted by the token site at scheduled times separated by τ_t seconds. In addition, T-RMP is a bulk acknowledgment protocol. An acknowledgment message contains a list of all of the source message that the token site has received, but which have not been acknowledged by the previous token sites. The t^{th} token passing message acknowledges a sequence of k source messages, where k is variable. The messages are assigned sequence numbers $s+1$ to $s+k$, where s is the last sequence number assigned in the $(t-1)^{th}$ acknowledgment.

In T-RMP we assume that the receivers have synchronized clocks. Synchronization may be performed on the multicast network using other protocols [7, 8, 9] or may be performed on a parallel network, such as a satellite network, with deterministic delays. The clock synchronization technique is not part of T-RMP and is not considered in this presentation.

The primary advantage of the timed protocol is that a receiver detects a missing token based upon the time that it was scheduled to be transmitted, rather than later events that occur at undetermined times in the future. Negative acknowledgments have much more significance in the scheduled protocol than in the event driven protocol.

In the event driven protocol, RMP, we cannot assume that a receiver that has not sent

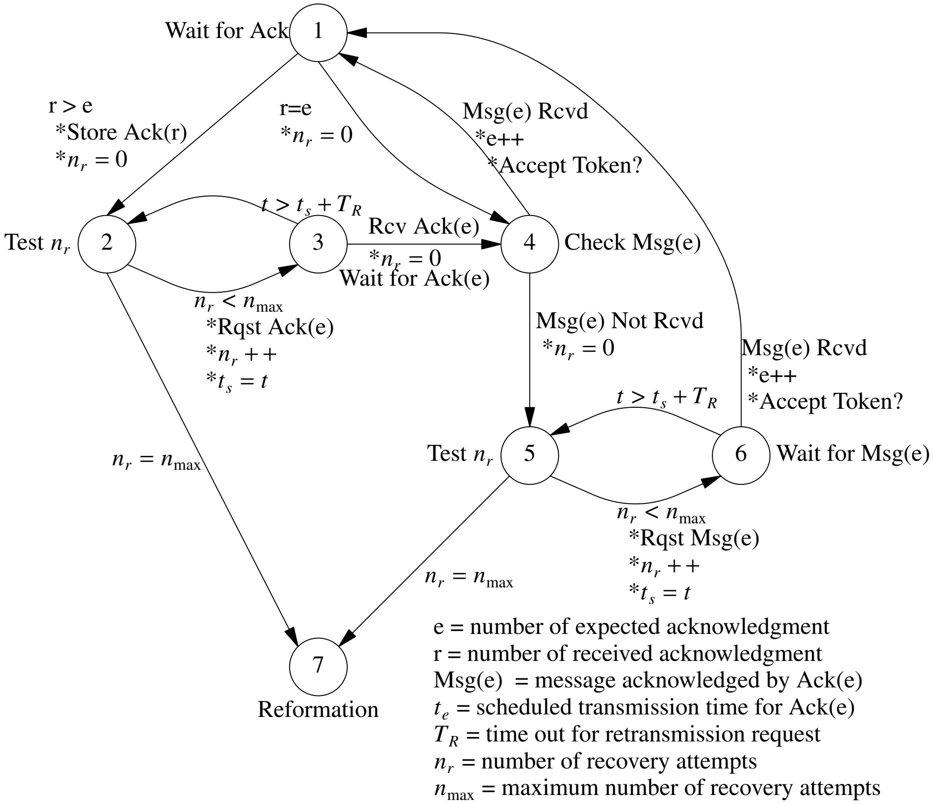


Fig. 2. An E-FSM representation of acknowledgment processing at a receiver in the RMP protocol

a negative acknowledgment has received a source message. The receiver may also have missed the positive acknowledgment for that source message and any subsequent acknowledgments that would indicate that it missed the first acknowledgment. We cannot be certain that the receiver has a source message until that receiver sends an implicit acknowledgment by sending a positive acknowledgment for a subsequent message.

In the scheduled protocol, T-RMP, a receiver is aware that it has missed an acknowledgment one network delay time after the acknowledgment is scheduled. Message recovery uses a positive acknowledgment protocol that retransmits unanswered requests at fixed intervals and declares a failure and places the system in reformation after a fixed number of unanswered requests. Therefore, after a fixed time following a message's acknowledgment, either all of the operable receivers have the message or the system is in reformation.

Figure 3 is the E-FSM representation of how acknowledgments are processed in T-RMP. We can use this state diagram to prove that all of the operable receivers have received a source message, or have placed the system in reformation within time $(n_{\max} + 1/2)T_R$ of when it was scheduled to be acknowledged. If the token site, that was scheduled to send the acknowledgment has failed, the system is placed in the reformation phase by the receivers. The sources don't have to detect a failed token site.

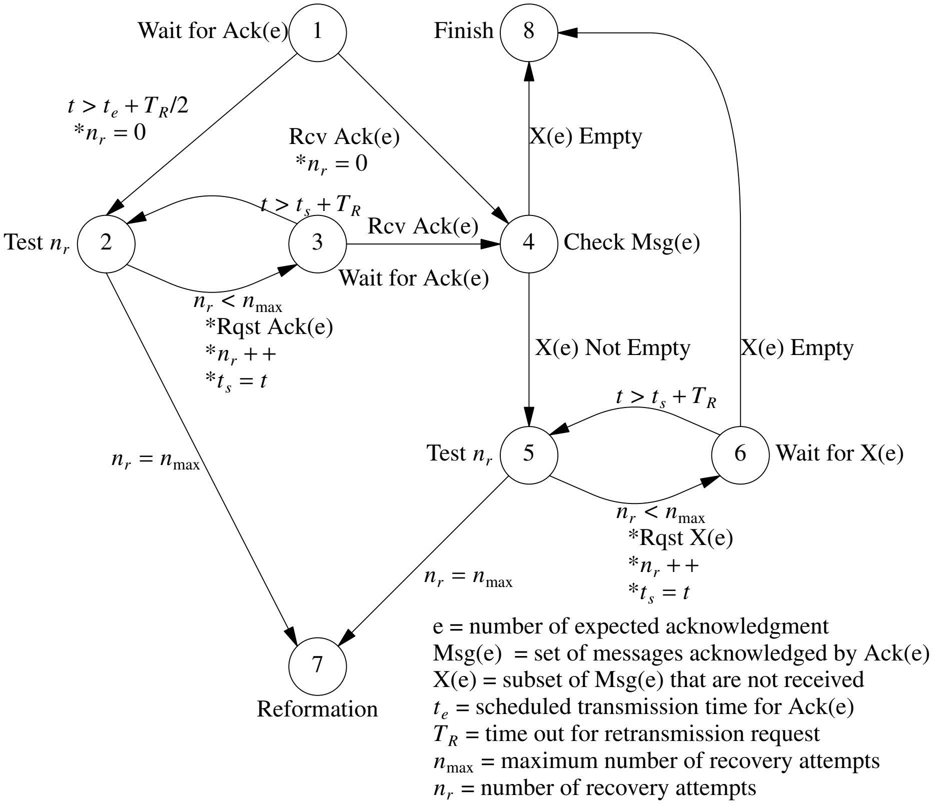


Fig. 3. An extended finite state machine representation of acknowledgment processing at a receiver in the timed RMP protocol

A source message is scheduled to be acknowledged at time t_e. If the acknowledgment is received before $t_e + T_R/2$, the receiver moves to state 4, with $n_r = 0$. Otherwise, at $t_e + T_R/2$ the receiver moves to state 2 with $n_r = 0$, requests the missing acknowledgment, increments n_r to 1, and moves to state 3. If the missing acknowledgment is received within T_R seconds, the receiver moves to state 4, otherwise it returns to state 2. The receiver circulates around the loop between states 2 and 3 at most $n_{\max}$ times. Either the receiver enters state 4 before

$t_e + (n_{\max} + 1/2)T_R$, or enters state 7, and initiates a reformation at time $t_e + (n_{\max} + 1/2)T_R$.

If a receiver enters state 4 at time t_4 such that $t_e + (k_4 - 1/2)T_R \leq t_4 < t_e + (k_4 + 1/2)T_R$, then $n_r = k_4$. If the receiver has the acknowledged source message, then the receiver move to state 8, otherwise it moves to state 5. If $k = n_{\max}$, the receiver moves immediately to state 7, otherwise it follows the 5->6->5 recovery loop up to $n_{\max} - k$ times. The receiver enters state 7 at time $t_4 + (n_{\max} - k)T_R < t_e + (n_{\max} + 1/2)T_R$, if it does not enter state 8 prior to this time. Therefore, by $t_e + (n_{\max} + 1/2)T_R$ all operable receivers either have the message, or have started a reformation process. If the token passing period is $T_P \geq (n_{\max} + 1/2)T_R$, the next token site has recovered all of the messages, and is ready to acknowledge messages before the next acknowledgment is scheduled to be transmitted.

The structure of the state machine for T-RMP is similar to the state machine for RMP in figure 2. Two obvious differences are that:

1. T-RMP moves from state 1 to state 2 when the local clock exceeds the scheduled acknowledgment time plus a reasonable network delay, while RMP makes the same transition when it receives a token with a larger sequence number than expected, and,
2. T-RMP checks for, and may have to recover, a set of source messages for each acknowledgment, while RMP only checks for a single source message.

There are two other things that should be noted in the T-RMP state machine,

1. the time out that activates the transition from state 1 to state 2 is half the time out that activates the transitions between states 2 and 3 or 5 and 6, and,
2. the sum of retries to recover a missing acknowledgment and a missing message, is limited, rather than separately limiting the number of retries to recover each.

The sum of the timer delays in T-RMP determine how frequently we can transfer the token. The smaller the timers, the more frequently we can transfer the token. The more frequently we are able to transfer the token, the smaller the time until we are certain that all of the receivers have a message. In addition, smaller token transfer times result in a smaller waiting time until source messages are acknowledged. Therefore, we would like to make the total timer delays as small as possible.

4 Merged Retry Count

We merge the count of retry requests to recover lost acknowledgments and lost messages because it reduces the maximum time that we allow to recover messages, without increasing the probability of erroneously entering the reformation phase. As an example, consider a system with independent messages losses, P_L:

— The probability that a receiver does not receive an acknowledgment is $P_A = P_L$;

— The probability that the receiver misses at least one of k source messages that are covered by an acknowledgment is $P_S = 1 - (1 - P_L)^k$;

— And, the probability that the request for a retransmission from a receiver, or the retransmitted acknowledgment message, or retransmitted source messages, is lost is $P_R = 1 - (1 - P_L)^2$.

In a system that allows n_1 attempts to recover a missing acknowledgment and a separate n_1 attempts to recover any missing source messages, the probability of initiating a reformation process because a sequence of messages has been lost, rather than because a component has failed, is

$$P_{R,1}(n_1) = (P_A + P_S)P_R^{n_1} - P_A P_S P_R^{2n_1} \ .$$

In a system that allow a total of n_2 attempts to recover both the missing acknowledgments and retries, the probability of initiating the same erroneous reformation is

$$P_{R,2}(n_2) = (P_A + P_S)P_R^{n_2} + P_A P_S\left(n_2 P_R^{n_2 - 1}(1 - P_{R)} - P_R^{n_2} \right).$$

When $P_L \ll 1$, using a Taylor series expansion,

$$P_{R,1}(n_1) \approx \frac{k+1}{2}(2P_L)^{n_1+1} \text{ , and,}$$
$$P_{R,2}(n_2) \approx \left(\frac{k+1}{2} + \frac{kn_2}{4} \right)(2P_L)^{n_2+1} \ .$$

For $n_2 P < 1$, which is reasonable considering that $P_L \ll 1$,

$$P_{R,2}(n_1) > P_{R,1}(n_1) > P_{R,2}(n_1 + 1)$$

In other words, if we make the sum of the retries one greater than the number of separate retries to recover the acknowledgment and source messages, we are less likely to initiate an erroneous reformation process. A system that allows 3 separate tries to recover acknowledgments and source messages must allow 6 recovery intervals before passing the token. A system that monitors the sum of the retries can provide better performance while only allowing 4 recovery intervals before passing the token.

Of course we can make the above model more accurate by

— allowing different loss probabilities for different length messages, a short acknowledgment message versus up to k source messages,

— considering time correlation of the losses, and

— taking into account other receivers that may miss the same messages.

Our objective, however, is to demonstrate the advantage of summing the retry attempts, rather than to recommend a specific number of attempts for a particular network condition. In a real network the loss and delay change continuously. We recommend increasing n_2 by one, and slowing down the token passing, when the receivers initiate unnecessary reformations, and decreasing n_2, and speeding up the token passing, when there is a long time between erroneous reformations. How long is long depends upon how badly we want to avoid erroneous reformations.

5 Separating Events

At each time t_e acknowledgment $Ack(e)$ is scheduled to be transmitted. $\Delta_T = t_{e+1} - t_e$ is the token passing period. Let $Ack(e)$ and the source messages that it acknowledges comprise the message set $Msg(e)$. At $t_e + \Delta_C$ all of the receivers that have recovered the source messages in $Msg(e)$ commit those messages. We assume that at $t_e + \Delta_C$ most, if not all, of the receivers have these messages. At $t_e + \Delta_R$ any receiver that has not recovered $Msg(e)$ initiates a reformation.

In our initial description of T-RMP $\Delta_T = \Delta_C = \Delta_R = \Delta_{init}$. This simplified the description and understanding of the protocol because the operation of the protocol is the same at every receiver and token site during every token passing interval. At each t_e, if the system is not being reformed all of the receivers, including the token site, have all of the $Msg(i)$ for *all* $i < e$. At t_e the token site transmits $Ack(e)$. At $t_e + T_R/2$ all of the receivers that do not receive $Ack(e)$ try to recover it. At $t_e + \Delta_R (\leq t_{e+1})$ any receiver that has not recovered $Msg(e)$ initiates a reformation process. Therefore, if the system is not being reformed, the operation at t_{e+1} is the same as the operation at t_e. In addition, t_{e+1} is the commit time for the messages acknowledged at t_e, since we can guarantee that all of the receivers have those messages.

When a source message is received at the token site it may wait up to Δ_T before the token is transmitted, and then must wait an additional Δ_C before the receivers commit the acknowledged message. We would like to make $\Delta_{\max} = \Delta_C + \Delta_T$ as small as reasonable, in order to provide stronger quality of service guarantees. In the initial system $\Delta_{max,init} = 2 * \Delta_{init}$. In this section we set $\Delta_T < \Delta_{init}$. However, in order to keep the probability that a receiver has a message the same as in the initial system, we must make $\Delta_C > \Delta_{init}$. We show that $\Delta_{\max} = \Delta_T + \Delta_C < \Delta_{max,init}$, for a certain range of Δ_T. We further reduce $\Delta_{\max}$ by making $\Delta_C < \Delta_R$. We justify this reduction by noting that false alarms, that cause unnecessary reformations, are generally more costly than the late arrival of a message.

When we make $\Delta_T < \Delta_R$ $Msg(e)$ may be recovered after $Ack(e+1)$ is scheduled to be transmitted, since $t_{e+1} < t_e + \Delta_R$. Recovering $Msg(e)$ after t_{e+1} does not have to affect the operation of a receiver that is not also the token site. The receiver can start recovering the missing components of $Msg(e+1)$ at the scheduled time whether or not is has completed the recovery of any $Msg(i)$, $i < e+1$. A receiver may have several recovery processes in progress simultaneously, or, since all of the requests for missing messages are directed to the current token site, the receiver may combine all of the requests into a single message.

However, when the site that is scheduled to transmit $Ack(e+1)$ fails to recover $Msg(e)$ before t_{e+1}, all of the receivers are affected. By the conventions of the protocol, the token site does not transmit an acknowledgment until it recovers the earlier messages and can service all retransmission requests. All of the other receivers may start transmitting their retransmit requests at $t_{e+1} + T_R/2$, but the recovery cannot start in earnest until after the token site completes its recovery and transmits the acknowledgment. In our model, the number of retries needed to recover messages, and the distribution of the recovery time, is independent of when the recovery starts. Therefore, if the recovery starts later than $t_{e+1} + T_R/2$, it will end later. If we make $\Delta_T < \Delta_{init}$, we must make $\Delta_R > \Delta_{init}$ order to keep the probability of reformation when there isn't a failure the same.

The operation of the token sites can be mapped onto the operation of a D/G/1 queue, where the period of the arrival process is Δ_T and the service process is the distribution of times to recover $Msg(e)$. In order to perform this mapping, site s_e, that transmits $Ack(e)$, arrives in the queue at time t_e. the scheduled time to transmit the acknowledgment. If site s_{e-1}, that transmits $Ack(e-1)$, has successfully transmitted $Msg(e-1)$ to s_e (that is to say, s_e has successfully recovered $Msg(e-1)$) before t_e, then the queue is empty, and immediately begins to service $Msg(e)$. The service time of $Msg(e)$ is the time needed to successfully transmit $Ack(e)$ from site s_e to site s_{e+1}, which is responsible for transmitting $Ack(e+1)$, and for s_{e+1} to recover any missing source messages in $Msg(e)$. If $Msg(e-1)$ is not transferred to s_e by t_e, s_e must wait for the transfer to be complete before beginning to service $Msg(e)$. Site s_e receiving the token at $t_e + \delta$ and beginning the next token transfer is equivalent to s_e arriving at the queue at t_e, and waiting until the previous service is completed at $t_e + \delta$ to begin its own service. Note that s_{e+1} begins trying to recover $Msg(e)$ at $t_e + T_{R/2}$, and combines any other missing messages with this request. This makes the service time independent of the past history of site s_{e+1}. Whenever s_e transmits the acknowledgment, s_{e+1} is ready to start recovery, without waiting for an earlier recovery to be complete.

The queue builds up because of the token passing process, but the waiting time distribution for the queue is the waiting time component for the delay at any receiver. None of the receivers can start recovering $Msg(e)$ until s_e has the token. Therefore they all have the same waiting time. The delay between the time that a source message is scheduled to be acknowledged and the time that a receiver has that message is the convolution of the waiting time distribution with the service time distribution. The service time distribution is the time needed for the receiver to acquire a message set $Msg(e)$, when the token sites have not failed. When the delay at a receiver reaches Δ_R, the receiver starts a reformation process, even though there has not been a failure. The waiting time is zero after a reformation. Since the probability of a false reformation is intentionally small, we approximate this probability as the probability of exceeding Δ_R in an infinite queue. The probability that a receiver has not acquired a source message when it is scheduled to be committed is the probability that the delay exceeds Δ_C.

Following the model in the previous section, the service time is

$$s = d_{n,1}(1 - x_A) + \frac{T_R}{2}\, x_A + j_1 T_R + d_{n,2} + d_{n,3} x_A + j_2 T_R + d_{n,4} + d_{n,5} x_S$$

where:
$d_{n,i}$ are delays through the network that depend on the source, the current token site and the network congestion,

$$x_A = \begin{cases} 1 & \text{with probability } P_L \\ 0 & \text{otherwise} \end{cases}$$

$$x_S = \begin{cases} 1 & \text{with probability } 1 - (1 - P_L)^r \\ 0 & \text{otherwise} \end{cases}$$

r is the number of arrivals from independent sources during Δ_T and is distributed as

$$p(r) = \frac{(\mu_A \Delta_T)^r e^{-\mu_A \Delta_T}}{r!}$$

and, j_i are the number of unsuccessful retransmission attempts before acquiring a missing message and is distributed as

$$p(j) = (1 - P_R) P_R^j \text{ for } j = 0, 1, 2, \ldots \text{ where } P_R = 1 - (1 - P_L)^2.$$

When the delay and retries are uncorrelated, the average service time is:

$$\mu_S = \mu_N \left\{ 1 + P_L + 2(1 - e^{-\mu_A \Delta_T P_L}) \right\} + \frac{T_R}{2} P_L + T_R \frac{P_R}{1 - P_R} \left\{ P_L + 1 - e^{-\mu_A \Delta_T P_L} \right\},$$

where $\mu_N = E(d_{N,j})$. When $P_L \ll 1$ and $\mu_A \Delta_T P_L \ll 1$,

$$\mu_S \approx \mu_N + P_L \left\{ \frac{T_R}{2} + \mu_N (1 + \mu_A \Delta_T) \right\}.$$

It's interesting to note that the time that it takes to transfer the token, s is a function of the token transfer period Δ_T and that μ_S decreases as the token transfer rate increases. When we transfer the token more frequently, fewer source messages arrive between token transfers, and it is more likely that we have not lost one or more messages. Therefore, when we transfer the token more often we are less likely to have to recover a source message. In the remainder of this section we are interested in the effect of decreasing Δ_T. In our first order analysis we will assume that μ_S is not a function of Δ_T. If we replace Δ_T with Δ_{init} ($\geq \Delta_T$), the value of μ_S will not decrease as we decrease Δ_T, and the actual advantage of decreasing Δ_T will be greater than we predict.

We do not know the service time distribution. The component of this distribution that is the distribution of network delay between the receivers and token site is difficult to determine and changes. We will use a negative exponential distribution for the service time. This is reasonable because of the memoryless property of the process that recovers lost messages. More importantly, this assumption replaces the D/G/1 queue with a D/M/1 queue, which we know something about. At this point we will make a number of approximations in order to get a feel for the quantitative relationship between Δ_T, Δ_C and Δ_R.

The waiting time distribution in a G/M/1 queue[10] is:

$$W(y) = 1 - \sigma e^{-(1-\sigma)y/\mu_S} \text{ for} y \geq 0, \text{where,}$$
$$\sigma = A^*((1-\sigma)/\mu_S),$$

and $A^*(s)$ is the La Place transform of the arrival time distribution.

In our case the arrival time distribution is deterministic with period Δ_T, so that

$$A^*(s) = e^{-s\Delta_T} \text{ and } \sigma = e^{-(1-\sigma)/\rho},$$

where $\rho = \mu_S/\Delta_T$. ρ is the utilization of the token passing channel. It if the fraction of the time the the token is in the process of being moved.

The equation for σ has one solution for $0 \leq \sigma < 1$, when $0 < \rho < 1$. This can be verified by considering the value of the exponential at $y = 0$ and $y = 1$, the slope at $y = 1$, and the second derivative over the range. The solution for sigma is plotted as a solid line in figure 4. Because of the shape of the curve, we approximate it with a quadratic. The least mean squared error fit is the quadratic:

$$\sigma = 1.168\rho^2 - .168\rho$$

The quadratic is plotted as the dashed line in figure 4. The fit is seen to be tight over the entire range.

The distribution of the delay is the convolution of the waiting time and service time distribution, and the probability of an erroneous reformation, P_{ref}, is the probability that the delay exceeds Δ_R. Therefore,

$$P_{ref} = e^{-(1+.168\rho-1.168\rho^2)\Delta_R/\mu_S}$$

In the initial system we expect the utilization of the token passing channel to be low because ρ is inversely proportional to Δ_T, $\Delta_T = \Delta_R$ and Δ_R is large enough that erroneous reformations occur infrequently. If ρ is small, the delay distribution is approximately equal to the service time distribution. This is a satisfying result for the initial system because whenever the waiting time is greater than zero, the system is put in reformation. The probability of reformation is approximately $P_{ref,init} \approx e^{-\Delta_{init}/\mu_S}$. The utilization is $\rho_{init} = \mu_S/\Delta_{init} = .43/\ln(P_{ref,init})$. If we adjust Δ_{init} so that $P_{ref,init} \leq 10^{-6}$, then $\rho_{init} \leq .07$, which justifies our claim that it is small.

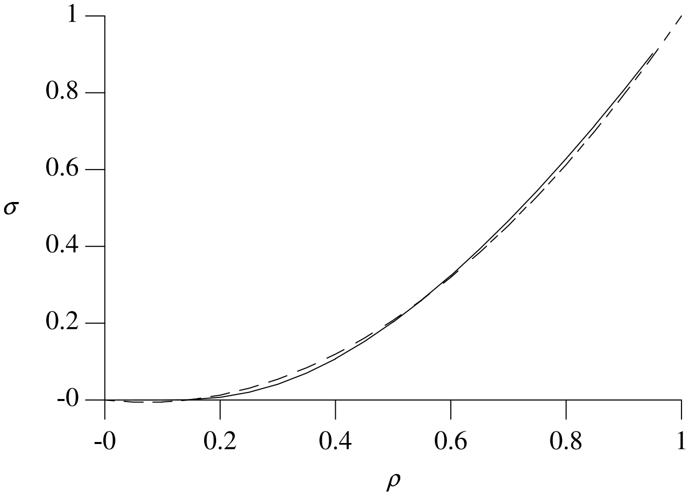


Fig. 4. Plot of $\sigma = e^{-(1-\sigma)/\rho}$ (solid curve) and $\sigma = 1.168\rho^2 - .168\rho$ (dashed curve)

Consider reducing $\Delta_T < \Delta_{init}$. ρ becomes larger. In order to keep P_{ref} the same, we must increase Δ_R so that

$$e^{-[1+.168(\mu_S/\Delta_T)+1.168(\mu_S/\Delta_T)^2]\Delta_R/\mu_S} = e^{-\Delta_R/\mu_S}.$$

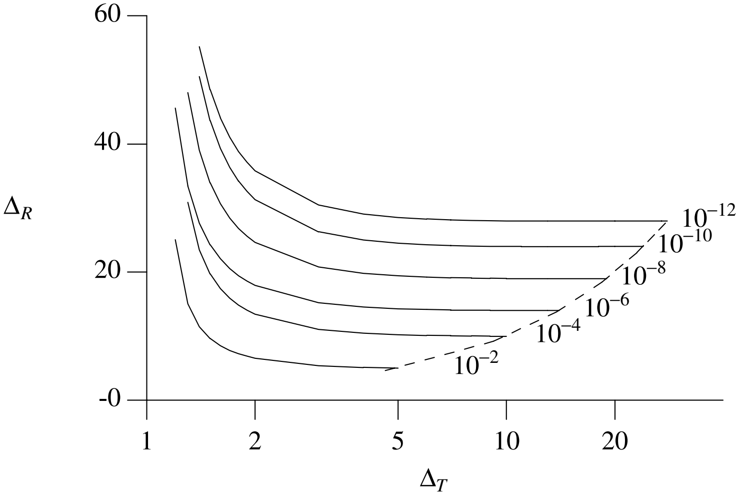


Fig. 5. Reformation delay Δ_R versus token passing period Δ_T for probabilities of erroneous reformation P_{ref} from 10^{-12} to 10^{-2}.

In figure 5 we plot equi-P_{ref} curves for P_{ref} from 10^{-2} to 10^{-12}. On the axes Δ_R and Δ_T are normalized with respect to μ_S, the average time that it takes to reliably transfer information between two participants in the protocol. The axis are dimensionless, and a value of 10 can be read as 10 average transfer times. The x-axis is also $1/\rho$. This axis is logarithmic with $\rho < 1$. The dashed curve is $\Delta_R = \Delta_T = \Delta_{init}$. We see that as we decrease Δ_T from Δ_{init}, Δ_R remains almost constant until ρ reaches about $30 - 60\%$, then grows rapidly as $\rho \rightarrow 1$. This graph shows us that there is almost no penalty for reducing Δ_T to $10 - 20\%$ of Δ_R.

Once a source message is received at the next token site, it may have to wait up to Δ_T until the next bulk acknowledgment message is scheduled to be transmitted, and then an additional Δ_C until the receivers use the message. The probability that a receiver has not acquired a message by Δ_C has the same form as the probability that the receiver has not acquired the message by Δ_R. The upper bound of this component of the message delay, $\Delta_T + \Delta_C$, normalized with respect to the message transfer time, is plotted in figure 6. The equi-probability lines are the probability that a receiver has not acquired the message by Δ_C. As Δ_T is reduced from Δ_{init}, the sum first decrease because Δ_C is increasing slowly. However, as $\rho \rightarrow 1$, Δ_C starts increasing quickly and the sum increases. There is a value of Δ_T that minimizes the sum, but the minimum is broad, so that the exact value of Δ_T is not critical.

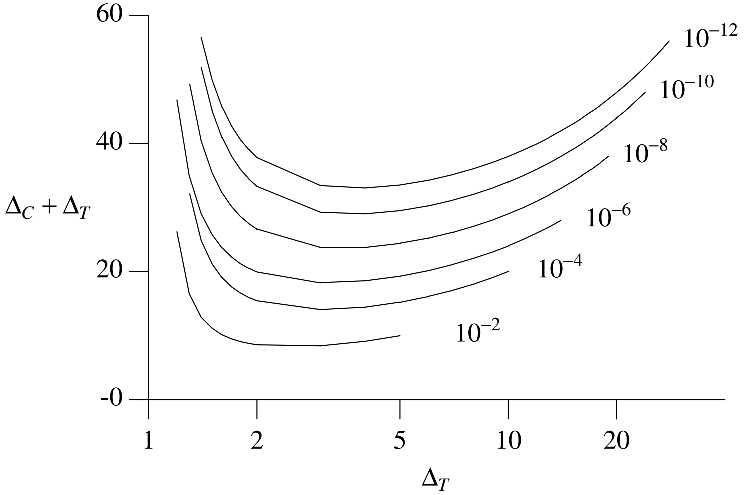


Fig. 6. Maximum delay from reception at token site to commit, $\Delta_C + \Delta_T$ versus token passing period Δ_T for probabilities that a receiver does not have a message by the commit time from 10^{-12} to 10^{-2}.

There are likely to be different penalties associated with a message arriving after the commit time and a system with the components operating properly entering a reformation. The quality of the information provided by a receiver may be adequate if 10^{-3} or 10^{-2} of the messages arrive after they are scheduled to be used. However, in a system with 100 receivers, if each receiver places the system in reformation with probability 10^{-2} each time the token is passed, the system will be placed in reformation after most token passes, and will spend most of its time in reformation. Therefore, Δ_C and Δ_R should be selected separately.

Suppose that the system can tolerate 10^{-4} of the messages arriving after they are scheduled to be used. From figure 6, the minimum of $\Delta_T + \Delta_C$ is approximately 14, and is achieved when Δ_T is about 3. If we also require that $P_{ref} = 10^{-10}$, from figure 5, the delay until we enter reformation, Δ_R, is about 26, only 2 greater than it was for Δ_{init}, as noted by the dashed line in figure 5. If we try to meet both constraints with $\Delta_C = \Delta_R$, $\Delta_T \approx 4$, $\Delta_T + \Delta_C \approx 29$, and $\Delta_R \approx 25$. The reformation delay improves by about 4%, but the component of the message delay more than doubles. Finally, if $\Delta_C = \Delta_R = \Delta_T$, as in the initial system, $\Delta_T + \Delta_C \approx 48$ and $\Delta_R \approx 24$. The message delay is about 3.5 times as large as it is in the system with independent selections, while the reformation delay improves by less than 8%. This example shows the importance of separating the selection of the delay.

6 Conclusion

We have shown that the guaranteed delivery delay in T-RMP can be reduced by combining the retry counters used to recover the token passing message and the missing source messages, and by using different time intervals to pass the token, Δ_T, commit messages, Δ_C, and to enter the reformation process, Δ_R. It is instructive to determine the reductions using reasonable numbers.

In original system $\Delta_T = \Delta_C = \Delta_R$, and the same number of retries n_r is allowed to recover the token passing message and the source messages. The maximum time from the reception of a source messages until it committed by all of the receivers is $\Delta_S = \Delta_T + \Delta_C$. In the initial system, $\Delta_T = (2n_r+.5)\Delta_N$, where n_r is the number of retries used to recover a missing message, and Δ_N is the nominal round trip network delay that we use to retransmit message recovery requests. The factor of 2 results from the two separate message recovery processes, and the factor .5 is the time that a receiver waits for the token passing message before initiating the recovery process. When the retry count for the two recovery processes are combined and $P_L \ll 1$, the total number of retries is limited to $n_t = n_r + 1$, so that $\Delta_T = (n_r + 1.5)\Delta_N$.

In typical ARQ protocols $n_r = 3$. Therefore, Δ_S in the combined system is $\frac{4.5}{7.5} = .6$ as large as in the original system. If the nominal round trip delay is one second, the maximum source delay is 13 seconds in the original system and 9 seconds in the combined system.

In the figures in section 5 all of the delays are normalized with respect to μ_s. If the selection of $n_r = 3$ and a nominal network delay of 1 second results in a probability of erroneous reformation $= 10^{-10}$, then, from the example at the end of section 5, $\Delta_T = \Delta_C = \Delta_R = 24\mu_s$. If we select the three periods independently, and allow 10^{-4} of the messages to arrive at some receivers after they have been committed by other receivers, then we can set $\Delta_T = 3\mu_S$, $\Delta_C = 11\mu_S$, and $\Delta_R = 28\mu_S$, and maintain $P_r = 10^{-10}$. This reduces Δ_S from 48 to 14, and the maximum source delay from 9 seconds to 2.625 seconds.

The two protocol modifications that we have studied provide a reduction in the delivery delay, in this example, of nearly 80%. It is worth noting that the guarantee is approaching the nominal network delay, so it is unlikely that further protocol modification will provide large improvements. In order to provide stronger delay guarantees we have to increase the number of retries in order to decrease the nominal network delay toward μ_s, or improve the operation of the network to reduce the actual network delay.

References

[1] A. N. Netravali, W. Roome, K. K. Sabnani, "Design and Implementation of a High-Speed Transport Protocol," IEEE Trans. Comm., vol. 38, no. 11, Nov. 1990, pp. 2010-2024.

[2] J-M. Chang, N. F. Maxemchuk, "Reliable Broadcast Protocols," ACM Transactions on Computer Systems, Vol. 2, No. 3, Aug. ’84, pp. 251-273.

[3] N. F. Maxemchuk, D. Shur, "An Internet Multicast System for the Stock Market," ACM TOCS, Aug. 2001.

[4] J-M. Chang, "Simplifying Distributed Database Systems Design by Using a Broadcast Network," Proc SIGMOD ’84, pp 223-233, June 1984.

[5] B. Whetten, G. Taskale, "An overview of reliable multicast transport protocol II," IEEE Network Mag., Jan/Feb 2000, pp 37-47.

[6] N. F. Maxemchuk, J-M. Chang, "Analysis of the Messages Transmitted in a Broadcast Protocol," Proc ICC ’84, pp 1263-1267, May, 1984.

[7] D. L. Mills, "Internet Time Synchronization: The Network Time Protocol," IEEE Trans. on Communications, Vol. 39, pp. 1482-1493, Oct. 1991.

[8] D. L. Mills, "Improved Algorithms for Synchronizing Computer Network Clocks," IEEE/ACM Trans. on Networking, Vol. 3, No. 3, pp. 245-254, June 1995.

[9] A. Ciuffoletti, "Uniform timing of a multi-cast service," Proc. of IEEE Conf. on Distributed Computing Systems, May 31- June 4, 1999. pp. 478-486.

[10] L. Kleinrock, **Queueing Systems -- Volume 1: Theory,** John Wiley & Sons, 1975.

Optimizing QoS-Based Multicast Routing in Wireless Networks: A Multi-objective Genetic Algorithmic Approach*

Abhishek Roy and Sajal K. Das

Center for Research in Wireless Mobility and Networking (CReWMaN)
Department of Computer Scienece and Engineering
University of Texas at Arlington
Arlington, Texas, 76019-0015, USA
{aroy,das}@cse.uta.edu

Abstract. With increasing demand for real-time services in next generation wireless networks, quality-of-service (QoS)-based routing offers significant challenges. Multimedia applications like video conferencing, real-time streaming of stock quotes or processing of scientific images relayed from satellites require strict QoS guarantee (e.g. bandwidth, delay) while communicating among multiple hosts. This gives rise to the need for an efficient *multicast routing* protocol which will be able to determine multicast routes satisfying the different QoS constraints. Design of such protocol boils down to a multi-objective optimization problem, which is computationally intractable. In fact, discovering optimal multicast routes is an NP-hard problem when the network state information is inaccurate – a common scenario in mobile wireless networks. In this paper, we propose a novel multicast tree selection algorithm that determines near-optimal multicast routes on demand. Based on the multi-objective genetic algorithmic (MOGA) approach, our solution attempts to optimize multiple QoS parameters (e.g. end-to-end delay, bandwidth guarantee and residual bandwidth utilization) simultaneously. We mathematically analyze the performance and convergence of the developed algorithm. Simulation results demonstrate that our algorithm is capable of discovering on-demand a set of QoS-based, near-optimal multicast routes within a few iterations, even with imprecise network information. From these set of routes one can choose the best possible multicast route depending on the specified QoS requirements.

1 Introduction

Multicast routing is an effective way to communicate among multiple hosts in a network. It outperforms the basic broadcast strategy by sharing resources along common links, while sending messages to a set of predefined destinations. This is particularly true in wireless networks which suffer from resource (bandwidth)

* This work was supported by NSF grants EIA-0086260, EIA-0115885, IIS-0121297 and Texas Telecommunication Engineering Consortium (TXTEC)

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 28–48, 2002.

scarcity and high bit error rate (BER). Furthermore, the growing demand for real-time multimedia communications like live video conferencing or streaming of stock quotes require strict *quality-of-service* (QoS) guarantee on such parameters as bandwidth, end-to-end delay, and delay jitter. An efficient allocation of network resources satisfying QoS requirements is the primary goal of QoS-based multicast routing [19]. However, individual QoS parameters may be conflicting and interdependent, thus making the problem even more challenging [15].

Further complications arise in wireless networks due to information (e.g. resource availability) inaccuracy caused by high BER and signal fading, leading to packet loss and hence higher packet delay and jitter. This effect can be reduced at the cost of extra bandwidth allocation. Thus, there exists a trade-off between bit error rate and bandwidth for a fixed radio spectrum. If we were to optimize a multicast route path with respect to a single QoS parameter, say bandwidth, then the problem can be solved in polynomial time even with uncertain network resources [8], by mapping it to a shortest path finding problem. On the otherhand, determining multicast routes satisfying different QoS parameters or constraints simultaneously, is an NP-hard problem [13]. The uncertainly of the network resources make such a problem more difficult. Therefore, various approximate algorithms have been proposed based on some heuristics.

Although QoS-Routing in wireless networks is an active research area in recent years, QoS-based multicasting is relatively a new research topic. The impact of information inaccuracy and uncertainty over QoS-routing has been investigated in [8,15] which proposes efficient heuristics to identify routes that are most likely to accommodate the desired QoS even with uncertain network state information. Using suitable probabilistic models it is shown that uncertainty is minimal for flows with only bandwidth requirements, but it makes path selections intractable when end-to-end delay is considered. A scalable, coarse-grained approach to control the mobile QoS is highlighted in [12]. The key technique used here is to aggregate a cluster of cells into a *Virtual Bottleneck cell* (VBC) in such a way that by controlling the parameters of VBC, specific QoS objectives of the system can be ensured without requiring the accurate prediction of the times and locations of each mobile user. The 3-level multi-agent architecture for QoS control in wireless ATM [11] provides a self-regulating network congestion control management by means of global network state awareness. A dynamic reconfiguration of the agents and an adaptive cell discarding scheme are performed to meet the end-to-end QoS requirements. The agents efficiently manage the buffer space to reduce the cell loss ratio while guaranteeing a bounded transit delay. In a completely different approach [16] multimedia streams are represented in terms of multiple substreams each with its own specified QoS and wireless network elements and protocols are made aware of the QoS requirements of such substreams. With the fluctuation of resource availability, using a fair scheduling algorithm the network selects and schedules substreams in order to meet an acceptable QoS. For effective multicast tree construction in interactive audiovisual communication, a heuristic has been proposed in [14] to compute low cost, delay-bound routes from source to each destination. Recently, the authors in

[1] demonstrated the efficiency of genetic-algorithm (GA) to obtain QoS-based multicast routes in computationally feasible time. With the help of evolutionary operations, the proposed algorithm is capable of optimizing multiple QoS parameters to generate a near-optimal multicast tree.

A careful analysis of the optimization schemes explored in QoS-routing in wireless as well as wirelined networks reveal that most of them suffer from the same drawback: multiple objectives are combined to form a *scalar single-objective* function on an ad hoc basis, usually through a linear combination (weighted sum) of multiple attributes. In these cases the solution not only becomes highly sensitive to the weight vector but also demands the user to have certain knowledge (e.g. priority of a particular objective, influence of an objective parameter over other) about the problem. Moreover, in the case of multi-objective optimization, a unique solution that optimizes all the objectives simultaneously will rarely, if at all, exist in practice. The user will therefore be more interested in obtaining a set of acceptable *non-dominated* solutions, one of which can be selected based on the specific problem requirements. We recognize that genetic algorithms can be readily modified to deal with multiple objectives by incorporating the concept of *Pareto-domination* (discussed in Section 2) in its selection operation [6].

In this paper, we use a *multi-objective genetic algorithm* (MOGA) technique to develop an efficient algorithm which determines multicast routes on-demand by simultaneously optimizing end-to-end delay guarantee, bandwidth requirements and residual bandwidth utilization without combining them into a single scalar objective function. Using suitable genetic operators, the algorithm is capable of finding near-optimal solutions within a few iterations. We have shown that with the increase in the number of nodes our algorithm performs better than existing algorithms based on scalar optimization. Although, it is impossible to provide a tight-bound for convergence of such an NP-hard algorithm, we have shown that asymptotically it can converge to the optimal point. From the experimental results it is clear that our algorithm is capable of obtaining more than 95% of the global optimal values for all three QoS parameters.

Section 2 reviews the basic concept of MOGA relevant in this context. The formulation of the required optimization functions and the proposed new algorithm are presented in Section 3. Section 4 highlights the power of the algorithm by analyzing the variation of some genetic operators and demonstrating its asymptotic convergence. In an attempt to evaluate the performance and the of the algorithm, a suitable model is developed and steady-state probabilities are calculated in Section 5. Simulation results in Section 6 corroborates the fast optimization of the required QoS parameters. Section 7 concludes the paper with pointers to the areas of future work.

2 Evolutionary Algorithms in Multi-objective Optimizations

Genetic algorithms (GA) provides a *guided random* search and optimization technique, based on the basic principles of *evolution: survival of the fittest* and *inheritance* [7]. It uses *probabilistic transition rules* and a *payoff* function to guide the search. All generalized greedy and gradient descent search techniques suffer from getting stuck at a *local optimal* point. However, using the evolutionary techniques, GAs can overcome this limitation to provide a *near-optimal* solution in a few iterations. The steps involved in solving an optimization problem using GA can be briefly summarized as follows: (i) Random generation of a *population* of *chromosomes*, (ii) Decoding each chromosome to evaluate its *fitness*, (iii) Performing *selection, cross-over* and *mutation* operations, (iv) Repeating steps (ii) and (iii) until a stopping criterion is satisfied. To solve any optimization problem, GAs start with *chromosomal representation* of the parameter set. A set of such chromosomes or strings are termed as *population.* The *fitness/objective function* is chosen in such a way that the good points in the search space possess high fitness values. This is the so-called *payoff* information used by GAs. In short, GAs mimic the natural evolution process through its selection, cross-over and mutation operations as discussed below:

- **Selection**: The selection process copies parent strings into a tentative new population known as *mating pool.* Selection is usually proportional to an individual's fitness value and thus mimics the evolutionary selection process. *Roulette wheel* selection, *stochastic universal* selection and *tournament* based selections are the most widely used techniques [4].
- **Cross-over**: The key idea behind the cross-over is to exchange information between two randomly selected parent-strings to give birth to the offsprings for the next generation. The selected strings from the mating pool are paired at random and a particular *cross-over point* is selected uniformly at random between position 1 and the string-length. The offsprings are generated by swapping the respective portions of the strings after the cross-over point.
- **Mutation**: Mutation is the process of *random alteration* in the genetic structure to introduce *genetic diversity.* In adverse situation, when the global optimal solution resides in a particular portion of the search space not included in the population, then the mutation is the only way to direct the population to *jump out* from any *local optimal* solution by randomly altering the information in the string.

In addition to these basic concepts, generally the best string up to a particular generation is preserved in a location either within the population or outside it. This idea is known as *elitism* [7]. We are now in a position to digress into its multi-objective counterpart of GAs.

A careful look into many real world problems reveals the requirement of simultaneous optimizations of multiple objectives. In principle, multi-objective optimization is quite different from the single-objective optimization. In case of multiple objectives, there may not exist a single best solution with respect to all the objectives. In fact, there exists a set of solutions superior to the rest of the solutions in the entire search space when all objectives are considered. These solutions are termed as *Pareto-optimal solutions.* Since none of the solutions in this set is *absolutely* better than any other, any one of them will be an acceptable solution. Hence, the user is given the freedom to choose the best solution from this set of Pareto-optimal solutions, defined below, to conform to specific requirements.

Pareto-optimal Front: This concept of *Pareto-optimality*, originally formulated by V. Pareto in the 19th century and constitutes by the origin of research in multi-objective optimization. We can say that a point x is *Pareto optimal* if for every x either, $\cap_i(f_i(x) = f_i(x^*))$ or, there is at least one i such that $f_i(x) > f_i(x^*)$, $\forall i \in \mathbf{I}$, where $f_i(x)$ is the *fitness function*. In other words, x^* is Pareto optimal if there exists no feasible vector x which would decrease some criterion without causing a simultaneous increase in at least one other criterion.

The *multi-objective genetic algorithm* (MOGA) varies from the ordinary GAs only in its selection operator. Before the selection is performed, using some suitable ranking schemes, the population is ranked on the basis of individual chromosome's or string's *non-domination*. The non-dominated strings from the current population are first identified to form the first *Pareto-optimal* front [5]. As MOGA iterates in every generation, the non-dominated, *Pareto-optimal* solutions are found and genetic operations are performed on these solution-sets to improve their fitness values. The non-dominated solution sets quickly proceeds towards the global optimal solution and gets saturated at a near-optimal solution-set. However, the tournament selection method used for ranking schemes can lead to a tie between two or more strings which is resolved by *Niche sharing* discussed below.

Niche Sharing on Non-dominated Frontier: Fitness sharing has already been applied to a number of real world problems. Given an optimization function having several peaks, the goal of fitness sharing is to distribute the population over the different peaks in the search space, where each peak receives a fraction of the entire population according to its height. The easiest way to achieve such fitness-sharing is to degrade an individual's *fitness*, f_i, by dividing it by a *niche count*, m_i, for that individual. The intuition behind the niche count is that it is a good estimate about how *crowded* the *neighborhood* of a particular individual i is [9], [20].

With these discussions we will now proceed to develop the multicast routing algorithm required for our protocol.

3 QoS-Based Multicast Routing Algorithm

The primary goal behind designing this algorithm is to find optimal multicast routes satisfying the necessary (QoS) parameters. Let us first discuss the different objective functions that the algorithm should try to optimize.

3.1 Objective Functions

Since wireless networks often suffer from uncertainty of resources, we design the algorithm in such a way that it can determine the multicast routes by probabilistically satisfying three major objective parameters: (i) end-to-end delay requirement, (ii) bandwidth guarantee and (iii) residual bandwidth utilization.

We represent the network by a graph $G = (V, E)$ where V is the set of nodes and E is the set of edges between the node-pairs. A path between a source (v_s) and a particular destination (v_d) is represented by a sequence of nodes $v_s, v_1, v_2, v_3, ..., v_d$ where $v_i \in$ V. There can be multiple such paths between a given pair of source and destination. For *unicast* routing the problem is to find the most efficient path between such a given pair of source and destination satisfying the required QoS constraints. However, in *multicast* routings, our focus is to find such paths between a single source and multiple destinations,which will simultaneously satisfy the above QoS parameters. These multicast paths essentially forms a *multicast tree* and we have multiple such trees.

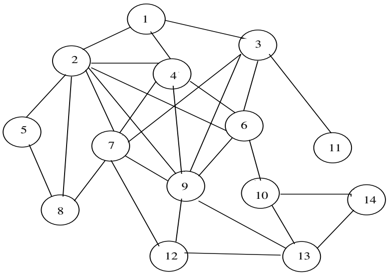


Fig. 1. A graph representing network

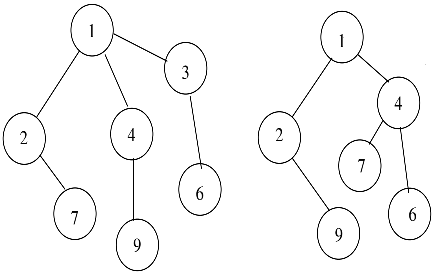


Fig. 2. Two different valid multicast trees

Figure 2 shows two possible multicast trees for finding routes from the source node 1 to destination nodes 6, 7, 9 for the input network of Figure 1. But, not all these paths can meet the desired QoS requirement. Our algorithm will look for the *set of non-dominated paths* that will satisfy the three different QoS parameters, namely end-to-end delay, bandwidth guarantee and residual bandwidth utilization. We assume the network to satisfy the following properties:

- The links are assumed to be *service queues* where packets are transmitted and get serviced. The *service rate* is assumed to follow *Poisson distribution* which makes the service time to obey *Exponential distribution.* The link delays introduced due to service time, should also follow an *Exponential distribution* with parameter equal to λ. Since the path consists of a chain of k hops, the delay along the entire path should follow *Erlang-K distribution* [17], which is the convolution of k independent random variables, each having the same exponential distribution. The probability that the delay (d_p) over a path p (from the source to one of the multicast destinations) of length k is less than t is given by: $Pr(d_p < t) = \frac{\lambda^k t^{k-1} e^{-\lambda t}}{(k-1)!}$. The probability that the delay (d) of the selected multicast tree $(\mathcal{T})$ will meet the specific delay constraint, can be obtained by taking the product of delays over individual paths in that multicast tree. This is expressed by: $Pr(d_{\mathcal{T}} < t) = \prod_{p \in \mathcal{T}} Pr(d_p < t)$. Our algorithm attempts to *maximize* this probability.

- To measure the second optimization factor, bandwidth guarantee, a similar model for the network links is assumed. Assume the *service* or *transmission rate*, a good measure of link bandwidth, follows a *Poisson distribution.* Then the probability that a link $l \in E$ is capable of providing a bandwidth of B is given by: $Pr_l(B) = \frac{\lambda^B e^{-\lambda}}{B!}$. The probability with which the bandwidth guarantee of B is satisfied for an entire multicast tree ($\mathcal{T}$) is given by: $Pr_{\mathcal{T}}(B) = \prod_{l \in \mathcal{T}} Pr_l(B)$. Our algorithm will try to *maximize* this probability also.
- Our third optimization factor is *residual bandwidth utilization.* Generally, the multicast path capable of providing *greatest residual bandwidth* is taken as the best possible choice. The total residual bandwidth in the network after allocating bandwidth for multicast is given by $\sum_{l \in E}(c_l - b_l)$, where c_l is the capacity of a link $l \in E$ and b_l is the bandwidth allocated for different hops along the multicast tree ($\mathcal{T}$). One can easily notice that $b_l = 0$ if $l \not\in p$, where $p \in \mathcal{T}$. The fraction of total bandwidth available as residual bandwidth is given by: $R_b(\mathcal{T}) = \frac{\sum_{l \in \mathcal{T}}(c_l - b_l)}{\sum_{l \in \mathcal{T}} c_l}$. This measure is the third objective function that our protocol should try to maximize.

  We have also taken the *call blocking rate* as the measure of performance to compare our protocol with other existing ones. In order to determine the number of blocked calls, we first estimate the minimum available bandwidth for the multicast tree as $b_{avail}^{min} = \min_{l \in \mathcal{T}}(b_{avail}^{l})$, where $b_{avail}^{l} = c_l - b_l$ is the residual bandwidth on a network link belonging to the multicast tree $\mathcal{T}$. Any multicast session request is considered as *blocked* if its bandwidth requirement is more than b_{avail}^{l}.

  We now proceed to develop an efficient algorithm for on-demand QoS multicasting.

3.2 Proposed Algorithm

The underlying concept of the algorithm in Figure 3 is that it does not combine the three QoS objective functions on an ad hoc basis to form a scalar objective function, but attempts to tackle the problem from the perspectives of multi-objective optimizations. The motivation behind developing such an algorithm is to provide the user with a set of *Pareto-optimal* solutions, and give the liberty to choose the best solution from the set, depending on the specific requirements. We now discuss the implementation details of our algorithm and highlight the basic flow in Figure 4.

3.3 Implementation Details

The detailed implementation of the algorithm is discussed below.

Line 1: The *Network-generation* part of the algorithm takes the *number of nodes* as input and dynamically generates the graph using adjacency matrix representation with random connectivity.

MOGA-based Multicast-routing Algorithm

```
1. Generate a network (input: number of nodes) with random connectivity;
2. Obtain the initial set of multicast trees (input: source, destinations);
3. Map each of the multicast tree to a string sequentially consisting network nodes;
4. Generate the initial population by taking a specific number of such strings;
5. Repeat
6.        Calculate the initial fitness values of three QoS parameters separately;
7.        Generate the comparison set ($\mathcal{C}$) from population;
8.        While (not all strings are examined)
9.               Take out two strings at random;
10.              Compare each of their fitness values with the strings in $\mathcal{C}$;
11.              If (one string dominates the other (considering all fitness values))
12.                 Mark the non-dominated string;
13.              End-If;
14.              If (tie occurs (i.e. both the strings are dominated / non-dominated ))
15.                 Calculate niche count;
16.                 Mark the string with lower niche count as non-dominated;
17.              End-If
18.              Add the non-dominated strings into Pareto-Optimal set ($\mathcal{S}$);
19.       End-While
20.       Perform cross-over and mutation operations;
21.       Obtain the new set of strings to get new population;
22. Until $(\{fitness\}_{\mathcal{S}_{new}} - \{fitness\}_{\mathcal{S}_{previous}} < \epsilon)$;
```

Fig. 3. Multi-Objective QoS-Multicasting Algorithm

Line 2: The algorithm now takes as input the *source node* v_s and a *specific number of multicast destination nodes*, say, $v_{d_1}, v_{d_2}, ..., v_{d_n}$ and finds a set of possible multicast paths from v_s to each of $v_{d_1}, v_{d_2}, ..., v_{d_n}$, using the *depth first search* (DFS) algorithm. This gives the initial set of multicast trees. Our goal is to find the multicast trees which will satisfy the required QoS parameters. The next step is to map the problem in a search space suitable to MOGA.

Lines 3-4: Each of the generated multicast trees is mapped to a string consisting of the sequence of nodes along the path from the source v_s to each of the destinations $v_{d_1}, v_{d_2}, ..., v_{d_n}$. To mark the end of a path from a source to a single destination, we use -1 as the *sentinel.* Figure 5 gives depicts this scenario where the second multicast tree of Figure 2 is represented by a string. The set of all such initial strings constitute the *initial population.*

Line 6: The *fitness_computation* computes the values of the three pre-defined QoS parameters individually. The objective of the algorithm now boils down to a search for different multicast paths which will improve the values of these QoS parameters at each iteration.

Line 7: The key idea behind developing *Pareto-optimization* is to use a *ranking selection* method to emphasize the good points and incorporate the concept of *niching* to maintain *stable subpopulations* of good points. In order to achieve good selection, a *comparison set*, of individuals are picked at random

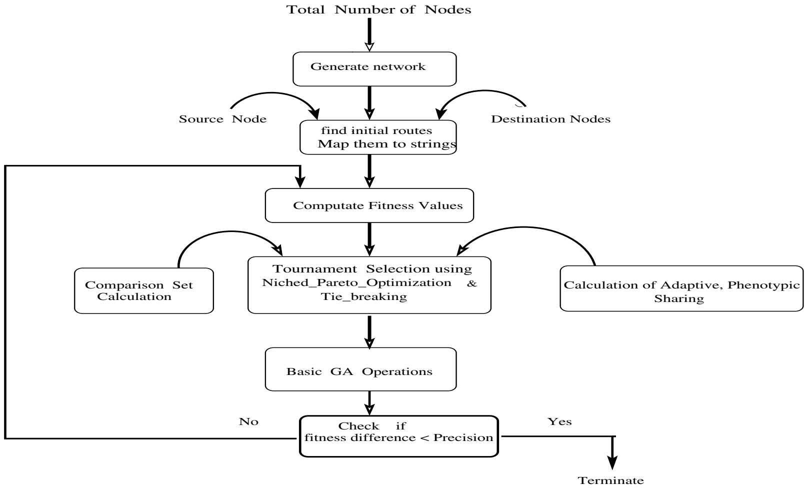

Total Number of Nodes
Generate network
Source Node
Destination Nodes
find initial routes
Map them to strings
Computate Fitness Values
Comparison Set Calculation
Tournament Selection using Niched_Pareto_Optimization & Tie_breaking
Calculation of Adaptive, Phenotypic Sharing
Basic GA Operations
No
Check if fitness difference < Precision
Yes
Terminate

Fig. 4. Flowchart of the Algorithm

1	2	7	−1	1	4	9	−1	1	3	6

Fig. 5. String representing the first Multicast tree of Figure 2

from the population. The size of this comparison set, t_{dom}, gives us a good control over the selection pressure. If a small t_{dom} is chosen, only a few Pareto-optimal points would be found. Instead, choosing a very large t_{dom} might result into a *premature convergence*. In this algorithm we have taken $t_{dom} = 0.20 \times$ (popsize).

Lines 8-16: From the population, two strings are randomly selected at a time and each of them is compared against each individual in the comparison set. If one candidate is dominated and the other is not then the latter is selected for selection. On the other hand, if both of the individuals are dominated or both non-dominated then we use *niche count* to resolve the tie. We compute the value of *niche count* for every individual string present in the population, is computed as:

$$m_i = \sum_{j=1}^{popsize} Sh[d_{s1,s2}], \tag{1}$$

where $d_{s1,s2}$is the distance between individuals $s1$ and $s2$ and $Sh[d_{s1,s2}]$ is the *sharing function*. For simplicity, triangular sharing function has been used:

$$Sh[d_{s1,s2}] = 1 - \frac{d_{s1,s2}}{\sigma_{share}} \tag{2}$$

for $d \leq \sigma_{share}$ and Sh[d] = 0 otherwise. Here σ_{share} is the *niche radius*, and it is a good estimate of *minimal separation* expected between the goal of solutions. Individuals within σ_{share} distance of each other degrade each other's fitness, as they are in the same niche. We introduce a new concept of *adaptive sharing*, i.e., the value of σ_{share} is no longer kept fixed. Depending on the fitness values of the particular string chosen and the population density in the search space, σ_{share} is dynamically updated in every iteration of the algorithm. We compute phenotypic Eucledian distance [9] between the different fitness values as a good measure of this σ_{share}.

$$d_{s1,s2} = \sqrt{(\delta_{delay_{s1,s2}})^2 + (\delta_{bw_{s1,s2}})^2 + (\delta_{bit_{s1,s2}})^2} \tag{3}$$

where $\delta_{delay_{s1,s2}} = Pr(d_{s1} < t) - Pr(d_{s2} < t)$, $\delta_{bw_{s1,s2}} = Pr_{s1}(B) - Pr_{s2}(B)$ and $\delta_{bit_{s1,s2}} = R_b(s1) - R_b(s2)$, B and R_b are the bandwidth and residual bandwidth respectively.

Similarly, we obtain the *niche radius*, σ_{share}, as some fraction (precisely half) of the maximum separation possible in the population, i.e.,

$$\sigma_{share} = \frac{\sqrt{(\delta_{delay_{max}})^2 + (\delta_{bw_{max}})^2 + (\delta_{bit_{max}})^2}}{2} \tag{4}$$

where $\delta_{delay_{max}} = Pr_{max}(d < t) - Pr_{min}(d < t)$, $\delta_{bw_{max}} = Pr_{max}(B) - Pr_{min}(B)$ and $\delta_{bit_{max}} = (R_b)_{max} - (R_b)_{min}$.

Lines 17-18: The cross-over and mutation operations are same as normal genetic algorithms. But a close look into the structure of the chromosome in Figure 5 reveals that these genetic operations can not be performed on any arbitrary gene (network nodes), as that may result in some illegal paths. Both the cross-over and mutation operations can only be performed at the end of an existing path, i.e., immediately after the particular *sentinel*, represented by -1. To give an equal probability to all such possible cross-over and mutation points, we randomly select one such point. To combine the good strings and simultaneously preserve the effective ones, we have taken the probability of cross-over as 0.7 and that of mutation as 0.1.

Loop 5-19: As the algorithm executes, at every iteration the genetic operations dynamically update the chromosomes (strings) and try to improve the corresponding probabilities until the difference of fitness values between the current Pareto-optimal set and the previous one is less than the precision ϵ.

3.4 Illustrative Example

Let us work out a small illustrative example to explain the essence of the algorithm, considering the network represented in Figure 1 with same source and destination nodes. The possible routes to nodes 6, 7 and 9 are respectively ($1 \rightarrow 2 \rightarrow 6$, $1 \rightarrow 3 \rightarrow 6$, $1 \rightarrow 4 \rightarrow 6$); ($1 \rightarrow 2 \rightarrow 7$, $1 \rightarrow 3 \rightarrow 7$, $1 \rightarrow 4 \rightarrow 7$); and ($1 \rightarrow 2 \rightarrow 9$, $1 \rightarrow 3 \rightarrow 9$, $1 \rightarrow 4 \rightarrow 9$). Thus, we have $3^3 = 27$ possible multicast trees. We take $(1 \to 2 \to 7,\ 1 \to 3 \to 6, 1 \to 4 \to 9)$; $(1 \to 2 \to 9,\ 1 \to 4 \to 7,\ 1 \to 4 \to 6)$; and $(1 \to 2 \to 6,\ 1 \to 4 \to 7,\ 1 \to 3 \to 9)$ as our initial multicast trees which will form the initial strings in the population set. The three QoS parameters $Pr(d_{\mathcal{T}} < t)$ for end-to-end delay, $Pr_{\mathcal{T}}(B)$ for bandwidth guarantee and $R_b(\mathcal{T})$ for residual bandwidth utilization are evaluated on this set and the QoS-based fitness values obtained are shown in table 1.

Table 1. Initial Multicast Trees with QoS Parameters

Initial multicast trees	$Pr(d_{\mathcal{T}} < t)$	$Pr_{\mathcal{T}}(B)$	$R_b(\mathcal{T})$
$1 \to 3 \to 6,\ 1 \to 2 \to 7,\ 1 \to 4 \to 9$	0.4×10^{-3}	0.003	0.54
$1 \to 4 \to 6,\ 1 \to 4 \to 7,\ 1 \to 2 \to 9$	0.3×10^{-3}	0.001	$R(\mathcal{T})$=0.52
$1 \to 2 \to 6,\ 1 \to 4 \to 7,\ 1 \to 3 \to 9$	0.1×10^{-3}	0.004	0.46

As the initial population is too small to generate an effective comparison set, probabilistically both string-1 and string-2 is initially included in this set. We pick out two strings (1 and 2) randomly from the initial population and compare them with the string in the comparison set. From the QoS parameters it is clear that string-1 dominates the string-2. Hence, string-1 is now included in the non-dominated set. In the next trial strings 2 and 3 are randomly picked up and the same procedure results in a tie as the fitness values indicate both of them as non-dominated. Hence, as discussed in the algorithm, we calculate the niche count using Equations (1), (2), (3), (4) and obtain $d_{s1,s2} = 0.0204$, $d_{s2,s3} = 0.0101$, $d_{s1,s3} = 0.08105$ and $\sigma_{share} = 0.0405$, which leads to $m_2 = 0.2375$ and $m_3 = 0$, as $m_3 > \sigma_{share}$. The lower niche count of string 3 includes it in the non-dominated, Pareto-optimal front. Since, all strings of the population are examined, we now exit from the while loop.

Since the probability of cross-over is quite high, it is performed over both the pairs of strings $1, 2$ and $2, 3$ by selecting the cross-over points at nodes 6 and 7 respectively. The resulting four new strings are : $(1 \to 3 \to 6,\ 1 \to 4 \to 7,\ 1 \to 2 \to 9)$, $(1 \to 4 \to 6,\ 1 \to 2 \to 7,\ 1 \to 4 \to 9)$, $(1 \to 4 \to 6,\ 1 \to 2 \to 7,\ 1 \to 3 \to 9)$, and $(1 \to 2 \to 6,\ 1 \to 4 \to 7,\ 1 \to 2 \to 9)$. On the contrary, as mutation is a rare event it has not occurred in the first iteration. The above process is repeated at every iteration until the improvement is less than our precision. We tabulate the QoS based non-dominated, Pareto-optimal solutions of every iteration in Table 2. Within 4 iterations the improvement of the Pareto-optimal set becomes less than the precision and we conclude that the algorithm has obtained a good solution. The final non-dominated set of multicast trees are shown in Figure 6. From Table 2 it is clear that no single multicast tree gives the best solution in terms of all three QoS parameters, but the first, second and third multicast tree gives the best probabilities for meeting end-to-end delay, residual bandwidth utilization and bandwidth guarantee respectively.

Complexity of the Algorithm: The genetic operators *cross-over* and *mutation* requires $O(n)$ time, where n is the total number of network nodes. Since, the genetic operations are performed on every string in the population, the com-

Table 2. Chart of Multicast Trees with QoS Parameters

Multicast Trees in Different Iterations	$Pr(d_{\mathcal{T}} < t)$	$Pr_{\mathcal{T}}(B)$	$R_b(\mathcal{T})$
$1 \to 3 \to 6,\ 1 \to 4 \to 7, 1 \to 2 \to 9$	0.4×10^{-3}	0.0045	0.58
$1 \to 4 \to 6,\ 1 \to 2 \to 7,\ 1 \to 4 \to 9$	0.6×10^{-3}	0.007	0.55
$1 \to 2 \to 6,\ 1 \to 2 \to 7,\ 1 \to 3 \to 9$	0.7×10^{-3}	0.005	0.575
$1 \to 3 \to 6,\ 1 \to 2 \to 7,\ 1 \to 4 \to 9$	0.55×10^{-3}	0.006	0.500
$1 \to 3 \to 6,\ 1 \to 2 \to 7,\ 1 \to 2 \to 9$	0.7×10^{-3}	0.006	0.565
$1 \to 4 \to 6,\ 1 \to 2 \to 7,\ 1 \to 2 \to 9$	0.5×10^{-3}	0.008	0.55
$1 \to 2 \to 6,\ 1 \to 2 \to 7,\ 1 \to 4 \to 9$	0.475×10^{-3}	0.0058	0.625
$1 \to 3 \to 6,\ 1 \to 3 \to 7,\ 1 \to 4 \to 9$	0.61×10^{-3}	0.0065	0.601
$1 \to 2 \to 6,\ 1 \to 2 \to 7,\ 1 \to 3 \to 9$	0.95×10^{-3}	0.008	0.645
$1 \to 4 \to 6,\ 1 \to 4 \to 7,\ 1 \to 3 \to 9$	0.775×10^{-3}	0.0095	0.635
$1 \to 4 \to 6,\ 1 \to 3 \to 7,\ 1 \to 4 \to 9$	0.821×10^{-3}	0.0081	0.665
$1 \to 3 \to 6,\ 1 \to 2 \to 7,\ 1 \to 4 \to 9$	0.95×10^{-3}	0.0082	0.641
$1 \to 4 \to 6,\ 1 \to 3 \to 7,\ 1 \to 3 \to 9$	0.90×10^{-3}	0.0090	0.667
$1 \to 2 \to 6,\ 1 \to 4 \to 7,\ 1 \to 3 \to 9$	0.88×10^{-3}	0.0097	0.655

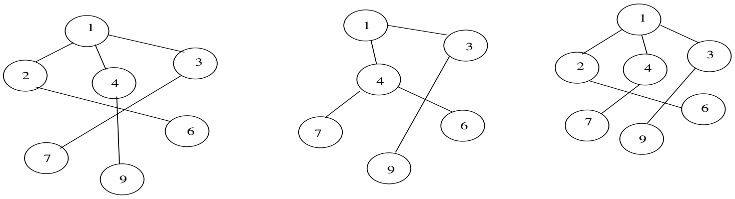


Fig. 6. Final Non-dominated Set of Multicast Trees

plexity of a single iteration of the algorithm will be: $O(\mathcal{P} \times n)$, where $\mathcal{P}$ is the population size. Finally, since, the algorithm is executed for g generations, the total complexity of the algorithm becomes $O(g \times \mathcal{P} \times n)$. The simulation experiments in Section 6 makes it clear that in most of the cases, only a few generations will give a near-optimal result. It is true that the number of iterations (g) varies with the population size ($\mathcal{P}$). A poor guess of choosing the initial population might increase the number of iterations leading to a relatively slower solution. However, such penalty is often tolerated while solving such a NP-hard problem.

Before going into the simulation results of the developed protocol, let analyze the algorithm to show its power, complexity and convergence.

4 Evolutionary Properties and Convergence

The general behavior of the algorithm depends on the fitness values of the individuals in the population. Using fitness distribution before and after the *selection* operation, several properties of the algorithm can be unveiled to show its power. But before proceeding further, we need to define the following distributions:

Cumulative Fitness distribution: *Fitness distribution* is a function that assigns to each fitness value $f_i \in \mathbf{R}$, the number of individuals in a population P carrying this fitness value. If $\eta \leq N$ is the number of unique fitness values and $f_1 < f_2 < ... < f_\eta$ is the ordering of the fitness values, then the *cumulative fitness distribution* $S(f_i)$ is the number of individuals with fitness value f_i or worse, i.e. $S(f_i) = \sum_{j=1}^{j=i} s(f_j)$, for $0 \leq i \leq \eta$.

Expected Fitness distribution: A *selection method* $\mathcal{M}$ is a function that transforms a fitness distribution s into another fitness distribution s' such that $s' = \mathcal{M}(s, parameter - list)$. The *expected fitness distribution* $\mathcal{M}^*$ after allowing a selection method s to the original fitness distribution ($\mathcal{M}$) is given by $\mathcal{M}^*(s, parameter - list) = E(\mathcal{M}(s, parameter - list))$, [7]. However, for simplicity, the notation s^* is often used to represent this expected fitness distribution. We will try to predict it out of a given distribution. In the selection process used in the algorithm, an individual with fitness f_i or worse can win the tournament if all other individuals have a fitness of f_i or worse. Hence, we need to calculate the probability that all other t individuals have worse fitness. As the probability to choose an individual with fitness f_i or worse is $\frac{S(f_i)}{N}$, we can say $S^*(f_i) = N\left(\frac{S(f_i)}{N}\right)^t$. Now, combining this with the relation $s^*(f_i) = S^*(f_i) - S^*(f_{i-1})$ from definition of cumulative fitness distribution, we get the expected fitness distribution on the multi-objective tournament selection process as:

$$s^*(f_i) = \mathcal{M}^*(s,t)(f_i) = N\left[\left(\frac{S(f_i)}{N})^t - (\frac{S(f_{i-1})}{N}\right)^t\right] \tag{5}$$

4.1 Analysis Using Continuous Distribution

We have assumed that the fitness values are continuously distributed. The continuous distribution $\bar{s}(f)$ will have the same range as its discrete counterpart. Hence, $\bar{S}(f) = \int_{f_0}^{f} \bar{s}(x)\partial x$ will be the expression for continuous cumulative distribution. We derive the probability of an individual with fitness f or worse to win the tournament as $\bar{S}^*(f) = N(\frac{\bar{S}(f)}{N})^t$. Again, as $\bar{s}^*(f) = \frac{\partial \bar{S}^*(f)}{\partial f}$, we obtain:

$$s^*(f_i) = \mathcal{M}^*(s,t)(f_i) = t\bar{s}(f)\left(\frac{\bar{S}(f)}{N}\right)^{t-1} \tag{6}$$

Selection Intensity: The *intensity* ($\mathcal{I}$) of the selection, defined as the expected average fitness value of the population after the iteration of the algorithm. Using normalized *Gaussian distribution* $G(0,1)(f) = \frac{1}{\sqrt{2\pi}}e^{\frac{-t^2}{2}}$ we have $\mathcal{I} = \int_{-\infty}^{\infty} f\bar{\mathcal{M}}^*(G(0,1))(f)\partial f$. Thus, the expression for selection intensity of our algorithm is given by

$$\mathcal{I}(t) = \int_{-\infty}^{\infty} tx \frac{1}{\sqrt{2\pi}} e^{\frac{-x^2}{2}} \left(\int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} e^{\frac{-y^2}{2}} \partial y \right)^{t-1} \partial x \tag{7}$$

We have varied the tournament-size t_{dom} and investigated the changes in $\mathcal{I}$. The plot in Figure 7 demonstrates that the intensity of selection increases with increasing tournament-size until the saturation arrives.

Selection Variance: The *selection variance* $\mathcal{V}$ is the expected variance of the fitness distribution of the strings after the algorithm completes its selection process over Gaussian distribution G(0,1). To calculate this variance with respect to our algorithm we evaluate the equation:

$$\mathcal{V}(t) = \int_{-\infty}^{\infty} t(x - I(t))^2 \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \left(\int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} e^{-\frac{y^2}{2}} \partial y \right)^{t-1} \partial x \tag{8}$$

Figure 8 shows the values of this selection variance with t_{dom}.

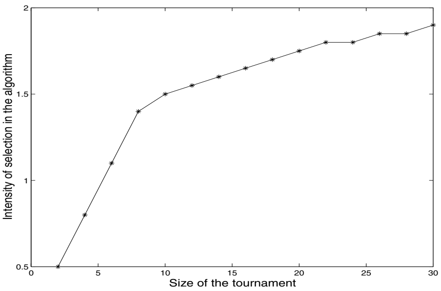


Fig. 7. Selection Intensity with respect to Tournament size

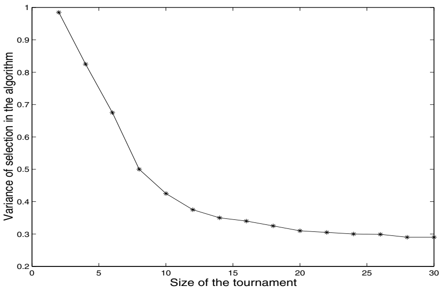


Fig. 8. Variance of the selection with respect to Tournament size

This provides us a trend of the selection pressure used in our algorithm. The selection pressure has its strong influence on selecting the good strings and punishing the bad ones, which eventually guides the improvement of the performance of our algorithm. Now, we will highlight on the convergence of the algorithm.

4.2 Convergence

While examining the convergence of the algorithm, we keep in mind that the proposed algorithm operates on the principle of *elitist GA*, i.e., in every iteration at least the current best individual strings survive. Intuitively, as the algorithm iterates, the fitness of the strings does not decrease. Let us assume that for every population P, there exists a non-zero probability Φ such that in the next generation the fitness of the population is better. Next, we divide the population into classes according to their fitness values. Suppose that the initial population has fitness value f_{init}, and the optimal fitness value is f_{opt}. Moreover, there are $r \geq 1$ intermediate fitness values. Also, let p denotes the minimum of all the probabilities $\Phi(P)$.

Now, we will proceed to give bounds for the probability that our algorithm reaches optimality in at most t iterations. In, the worst case this optimum will be obtained in exactly $t \geq r-1$ generations, if the $(r-1)^{th}$ improvement takes place in the t^{th} iteration. In order to realize this, we need to pick up $r-2$ different values from the set $\{1, 2, 3, ..., t-1\}$. Indeed, these numbers correspond to the steps where an improvement takes place. Here, we deal with the worst case scenario, in which improvements are as small as possible.

The lowest probability that the algorithm takes precisely t steps equals $p^{r-1}(1-p)^{t-r+1}\binom{t-1}{r-2}$, since we have $r-1$ improvements in the worst case, and $t-(r-1)$ times we get no improvement, i.e., the strings stay in the same class with probability $1-p$. So, the probability that we reach the optimum in at most $t \geq m-1$ steps is bounded by $p^{r-1}\sum_{i=r-1}^{t}(1-p)^{i-r+1}\binom{i-1}{r-2}$.

Using elementary calculus this sum equals:
$p^{r-1}\frac{1}{(r-2)!}\frac{\partial^{r-2}}{\partial q^{r-2}}\left(\sum_{i=1}^{t} q^{i-1}\right) = p^{r-1}\frac{1}{(r-2)!}\frac{\partial^{r-2}}{\partial q^{r-2}}\left(\frac{1-q^t}{1-q}\right)$, where $q = 1-p$. Differentiating and taking limits for $t \to \infty$ we get,

$$p^{r-1}\frac{1}{(r-2)!}\frac{\partial^{r-2}}{\partial q^{r-2}}\left(\frac{1}{1-q}\right) = p^{r-1}\left(\frac{1}{1-q}\right)^{r-1} = 1, \tag{9}$$

since $\frac{\partial^{r-2}}{\partial q^{r-2}}\left(\frac{1}{1-q}\right) \to 0$ as $t \to \infty$.

Therefore, we can conclude that the algorithm converges asymptotically to provide the optimum solution. In the next section we develop a suitable performance model for the proposed algorithm.

5 Performance Modeling Using Markov Chains

Markov chains can be used to model each generation of the algorithm by combining the effects of various stochastic events like initial population generation, selection, cross-over, mutation [18]. However, the major difficulty of it is that the transition probability matrix becomes large and unwieldy. To make the analysis simpler, we encode the node numbers in *binary* form to represent every string by 0s and 1s.

For a binary string encoded population of size $\mathcal{P}$ and M different states, a particular state i in the model represents a population with exactly i ones and $(\mathcal{P}-i)$ zeroes. The algorithm chooses a member k of the current population to reproduce with probability proportional to its fitness relative to total fitness of the population. Thus, leaving the effect of niche counts, we can choose an individual k with probability $\frac{f_k}{\sum f}$, where f_k is the fitness of k and $\sum f$ is the sum of the all individuals in the current population. Now, if f_1 and f_0 denotes the fitness of "1" and "0" respectively, then the probability p_1 of choosing a 1 for the next generation's population will be: $p_1 = \frac{i*f_1}{i*f_0+(\mathcal{P}-i)*f_0} = \frac{\hat{r}*i}{\hat{r}*i+(\mathcal{P}-i)}$, where

$\hat{r} = \frac{f_1}{f_0}$ is the fitness ratio. Similarly, the p_0 probability of choosing a zero will be: $p_0 = \frac{\mathcal{P}-i}{i*\hat{r}+(\mathcal{P}-i)}$. The probability of going from a state of i 1s to a state with j 1s will be: $p_{i,j} = \binom{\mathcal{P}}{j}(p_1)^j(p_0)^{\mathcal{P}-j}$. Substituting the values of p_1 and p_0, we get, $p_{i,j} = \binom{\mathcal{P}}{j}\left(\frac{\hat{r}*i}{\hat{r}*i+(\mathcal{P}-i)}\right)^j\left(\frac{\mathcal{P}-i}{\hat{r}*i+(\mathcal{P}-i)}\right)^{\mathcal{P}-j}$.

The above equation gives a probability transition matrix for the population size and the fitness ratio. However, the absence of niche counts is not incorporated in the equation. Hence, our next objective is to extend the equation to include the niche count into feature and model the algorithm exactly. As discussed earlier, the niche GA seeks to maintain several subpopulations, or individuals at different good solutions and it gives a good view of the fitness landscape. Each peak of such landscapes forms a niche. The sharing values now will modify the fitness values to spread the population out in different peaks. The niche counts for 1s, will be $m_1 = i + (\mathcal{P} - i)(1 - \frac{1}{\sigma_{share}})$, since we have to take care of the shared value of each zero. Similarly, the niche count of 0s will be $m_0 = (\mathcal{P} - i) + i(1 - \frac{1}{1-\sigma_{share}})$. The fitness values will also be changed to f_1/m_1 and f_0/m_0 respectively. Substituting these degraded fitness values to the previous equation, we get

$$p_{(i,j)} = \binom{\mathcal{P}}{j}\left(\frac{1}{1+\frac{(\mathcal{P}-i)(\mathcal{P}+\frac{i-\mathcal{P}}{\sigma_{share}})}{i\hat{r}(\mathcal{P}-\frac{i}{\sigma_{share}})}}\right)^j\left(\frac{1}{1+\frac{i\hat{r}(\mathcal{P}-\frac{i}{\sigma_{share}})}{(\mathcal{P}-i)(\mathcal{P}+\frac{i-\mathcal{P}}{\sigma_{share}})}}\right)^{\mathcal{P}-j} \tag{10}$$

This equation gives the probability of the transition matrix as the algorithm iterates from one state to another.

Absorbing Markov Chain: While calculating such transition probabilities, before talking about steady states, we need to address the absorbing states [3]. Although transition matrix will tell that the *quasi-steady* states can not last, we usually do not wait long enough to see that the algorithm has reached the equilibrium. We keep ourselves satisfied with just a *noisy steady state.* One possible way to deal with this problem is to ignore the steady states and just analyze only the transient states. Applying the well known partitioning of states of an absorbing Markov chain,we get: $P = \begin{pmatrix} QR \\ 0I \end{pmatrix}$.

We take only the Q partition to be the entire matrix, ignoring $R, 0, I$, which consists of only the absorbing states. If we normalize the Q matrix, the resulting matrix Q_{norm}, is an ergodic Markov chain that allows us to calculate the steady state probabilities for all non-absorbing states. Before analyzing Q_{norm}, we will try to justify the "chopping off" the absorbing states. Intuitively, we can say that we are only looking for the *expected absorption time.*

Ergodic Markov chain: We now have an irreducible Markov chain, Q_{norm}, with all ergodic states. Calculation of the steady-state probabilities is quite straightforward. We seek the steady-state probability-vectors $\overrightarrow{\Pi} = \{\pi_1, \pi_2, ..., \pi_{\mathcal{P}-1}\}$, where π_j denotes the steady-state probability for state j. To find $\overrightarrow{\Pi}$, we need to solve the equation $\overrightarrow{\Pi}Q = \overrightarrow{\Pi}$, where $\sum_{i=1}^{\mathcal{P}-1} \pi_i = 1$. Analyzing the vector $\overrightarrow{\Pi}$ helps us to understand the behavior of the steady state probabilities, plotted in Figure 9 against changing σ_{share} values.

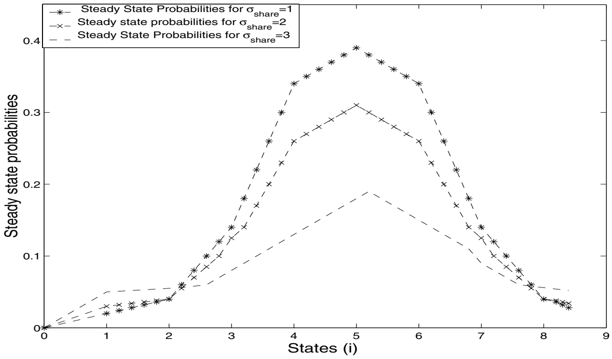


Fig. 9. Steady State Probabilities

All the steady state distribution curves are almost symmetric about the equilibrium point. As σ_{share} increases, the steady state distribution curve flattens and demonstrates the changing probabilities for different fitness sharing values.

6 Simulation Results

Simulation experiments are first performed over a network of $n = 100$ nodes with the number of multicast destination nodes being 10. The capacity of the network links are taken as uniformly distributed in the interval of [90-110]Mbps.Recall that our multicast QoS routing algorithm attempts to maximize the probabilities of meeting end-to-end delay, bandwidth requirement and bandwidth utilization within a few generations by building the Pareto- optimal fronts. We have compared the performance of our algorithm with an existing scalar-optimization [1] and heuristic algorithms [14] and observed that our algorithm performs better in terms of scalability and multicast call blocking rates.

An exhaustive search method, which finds the optimal values of the three QoS parameters by exhaustively searching them one after another is used to compare our results. The three plots (one for each QoS parameter) in Figures 10, 11 and 12 vividly explains how these Pareto-optimal fronts are developed and proceeded towards a global-optimal solution in a feasible time. The novelty of our algorithm is that it is capable of obtaining near-optimal values of all three QoS parameters simultaneously by building the non-dominated fronts. However, for the sake of clarity we have shown it in three different plots. Finally, after completing the execution of the algorithm, we get the final solution sets represented by the

Figure 13. From the above plots one can derive the amount of optimization obtained by our algorithm. Table 3 demonstrates that our algorithm is capable of obtaining more than 95% of the global-optimal values of end-to-end delay, bandwidth guarantee and residual bandwidth utilization within 100 iterations. As all the near-optimal solutions are achieved in a probabilistic approach, we conclude that our algorithm is robust enough to operate with imprecise network information. Note that the solution set may contain solutions which are not the best from any single objective's point of view, but is non-dominated by all three individual best solutions, when all three objectives are considered. Since the three individual best solutions will always be non-dominated, they are by default included.

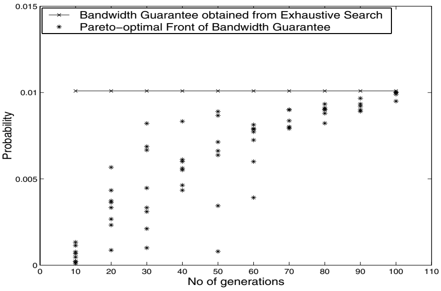


Fig. 10. Pareto-Optimal Bandwidth Guarantee with number of generations

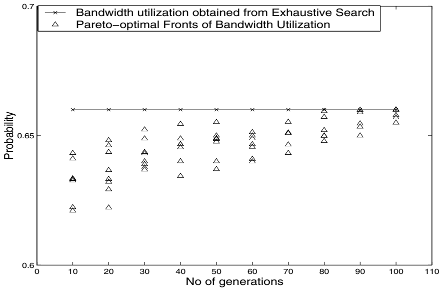


Fig. 11. Pareto-Optimal Bandwidth Utilization with number of generations

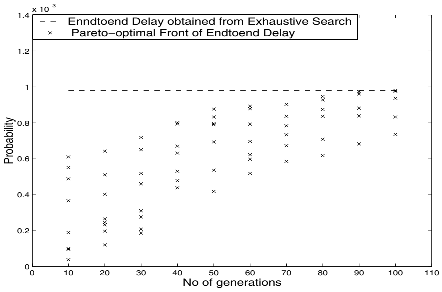


Fig. 12. Pareto-Optimal End-to-end Delay with no. of generations

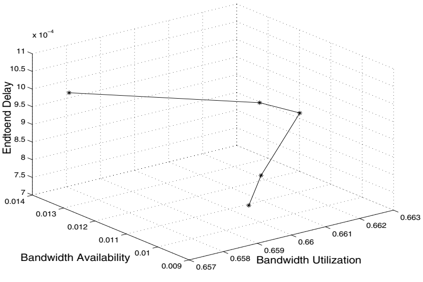


Fig. 13. Pareto-optimized Set of three QoS parameters

The solutions are provided in a generalized manner and a user can readily choose his choice-able solution depending on his needs. For example, in real time video transmission we are more careful about the end-to-end delay. Such a user

Table 3. Percentage of Global-optimal Solutions Obtained

End-to-end Delay	96.78
Bandwidth Guarantee	95.55
Bandwidth Utilization	98.39

will be having delay as a hard constraint and will choose the solution which will meet that constraint. On the other hand, while transmitting a scientific data from the remote satellite, the correctness is more important than the delay. So, for such cases an end-user will prefer to meet the bandwidth guarantee than the delay. We repeat the simulation with increasing number of network nodes and observe the efficiency of our algorithm. As the network becomes highly condensed, our algorithm exhibits a more linear and stable pattern than existing scalar optimization algorithm. This *approximate linearity* of the curve in Figure 14 corroborates the scalability of the algorithm. Finally, the non-dominated set of solutions are given as input to the call-blocking algorithm. Performance of our protocol is plotted against the increasing *call arrival rate* in Figure 15. The mean rate of arrival of multicast session request is assumed to be 10 requests/sec. Results show that the percentage of calls blocked in our protocol is less than the two existing QoS routing protocols based on scalar optimization [1] and heuristics [14]. The peak data rate for this comparison is taken as 35Mbps. Although the performance of all the schemes degrades with the increase of call arrival rate, our algorithm gains consistently over the existing ones. Hence, we can conclude that the designed protocol offers a *graceful degradation* of performance with increasing session arrival rates.

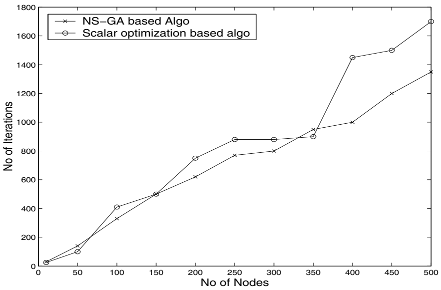


Fig. 14. Performance of the Algorithm with Increasing number of nodes

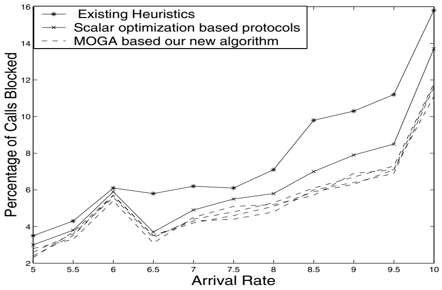


Fig. 15. Percentage of Calls Blocked with Call-Arrival Rates

7 Conclusions

On-demand multicast routing in networks is currently an active area of research. In most of the real world scenarios routings need to meet stringent measures of different quality of services. Seamless transmission of wireless audio and video traffic has already become a real challenge of current and future generation wireless systems. It is quite natural that real time multimedia traffic should meet a number of different and conflicting QoS issues. Optimizing a particular objective function may sacrifice optimization of another dependent and conflicting objective. In this paper, we studied QoS-based multicast routing problem from the perspective of multi-objective-optimizations. The blessing of multi-objective-genetic algorithms (MOGA) has paved the way to develop the algorithm for a new QoS-based multicast on-demand routing algorithm. The mathematical analysis shows the power of selection and complexity of the algorithm. We have also shown the asymptotic convergence of the algorithm to the optimal point. However, often we do not need to wait till the convergence and settle with a near optimal point. We have also developed a suitable model of the algorithm using Markov chains to track the transition probabilities and plot the steady state values of such probabilities. Simulation results delineates the efficiency, performance and scalability of the protocol. Our future interests is to adapt this technique to develop a mechanism for *renegotiable-QoS* in wireless multicasting. We expect our work will be helpful in solving some new problems in the domain of quality-of-service (QoS) routing.

References

1. N. Banerjee and S. K. Das, "Fast Determination of QoS-based Multicast Routes in Wireless Networks using Genetic Algorithms" *International Conference for Communication*, vol. 8, pp.2588-2592, 2001.
2. C. A. C. Coello, "An Updated Survey of GA-Based Multiobjective Optimization Techniques," *Technical Report, Laboratorio Nacional de Informatica Avanzada (Lania), Xalpa, Veracruz, Mexico*, pp. 1-45, June 1998.
3. J. N. Darroch and E. Seneta, "On quasi-stationary distributions in absorbing discrete-time finite Markov chains," *Journal of Applied Probability*, pp. 88-100, 1965.
4. L. Davis, "Handbook of Genetic Algorithms," *Van Nostrand Reinhold*, New York, 1991.
5. K. Deb and D. E. Goldberg "An investigation of niches and species formation in genetic function optimization," *Proceedings of the third International Conference on Genetic Algorithms*, pp. 42-50, 1991.
6. C. M. Fonesca and P. J. Fleming, "Genetic Algorithms for Multiobjective optimization: formulation, discussion and generalization," *Proceedings of the Fifth International Conference on Genetic Algorithms*, pp. 416-423, 1993.
7. D. E. Goldberg, " Genetic Algorithms : Search, Optimization and Machine Learning," *Adison-Wesley*, 1989.
8. R. A. Guerin and A. Orda, "QoS Routing in networks with Inaccurate Information: Theory and Algorithms," *IEEE/ACM Transactions on Networking*, vol. 7, no. 3, pp. 350-364, June 1999.

9. J. Horn, N. Nafpliotis and D. E. Goldberg, " A Niched Pareto Genetic Algorithm for Mutiobjective Optimization," *IEEE Conference on Evolutionary Computation*, New Jersey, vol. 1, pp. 82-87, 1994.
10. P. G. Harrison and N. M. Patel, "Performance modeling of Communication Networks and Computer Architectures," *Addison Wesley*, Reading, MA, 1989.
11. Y. Iraqi, R. Boutaba and A. Leon-Garcia, "QoS Control in Wireless ATM," *Mobile Networks and Applications*, vol. 5, pp. 137-145, 2000.
12. R. Jain, B. Sadeghi and E. W. Knightly, "Towards Coarse-Grained Mobile QoS," *Workshop on Wireless Mobile Multimedia*, pp. 109-116, 1999.
13. V. P. Kompella, J. C. Pasquale and G. C. Polyzos, "Multicasting for Multimedia Applications," *Proc. of IEEE Infocom 92*, Florence, Italy, vol.3, pp. 2078-2085, May 1992.
14. V. P. Kompella, J. C. Pasquale and G. C. Polyzos, "Multicast Routing for Multimedia Communications," *ACM/IEEE Transaction on Networking*, vol. 1, no. 3, pp. 286-292, Jun. 1993.
15. D. H. Lorenz and A. Orda, "QoS Routing in networks with Uncertain Parameters: Theory and Algorithms," *IEEE/ACM Transactions on Networking*, vol. 6, no. 6, pp. 768-778, Dec. 1998.
16. M. Naghshineh and M. W. Wilebeek-LeMair, "End-to-End QoS provisioning in Multimedia Wireless/Mobile Networks Using an Adaptive Framework," *IEEE Communications Magazine* vol. 6, no. 6, pp. 72-79, Nov. 1997.
17. R. Nelson, "Probability, Stochiastic Processes, and Queueing Theory," *Springer-Verlag*, 1995.
18. A. E. Nix and M. D. Vose, "Modeling Genetic Algorithms", *Analysis of Mathematics and Artificial Intelligence*, vol. 5, 1992.
19. S. Shenker, C. Patridge and R. Guerin, "Specification of Guaranteed Quality of Service, " Request for comments *RFC 2212, Internet Engineering Task Force*, Sept. 1997.
20. N. Srinivas and K. Deb, " Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms" *Journal of Evolutionary Computation*, vol. 2, no. 3, pp. 221-248, 1995.

An Experimental Study of Probing-Based Admission Control for DiffServ Architectures

Susana Sargento[1], Roger Salgado[1], Miguel Carmo[1], Victor Marques[2], Rui Valadas[1], and Edward Knightly[3]

[1] University of Aveiro/Institute of Telecommunications, 3810 Aveiro, Portugal,
susana@ua.pt, roger@av.it.pt, etmac@ua.pt and rv@ua.pt
[2] Portugal Telecom Inovação, 3810 Aveiro, Portugal,
victor-m-marques@ptinovacao.pt,
[3] ECE Dept., MS380, Rice University, Houston, TX 77005, USA,
knightly@ece.rice.edu

Abstract. Probing is a well-known admission control technique that can achieve high utilization and per-flow quality of service in a scalable way. We have recently introduced an extension to the basic probing technique, called ε-probing, to overcome a resource stealing problem that impairs the use of probing in systems with multiple service classes. In this paper we describe an experimental system that was designed to evaluate the effectiveness of both probing and ε-probing techniques. We have developed a software module that implements the probing functionality, which can be inserted in end hosts or edge routers. Several tests were carried out to study the effect of various system parameters in the performance of the probing techniques. The results clearly show that both probing techniques are able to accurately perform admission control while achieving high utilization. Moreover, they also show that in environments with multiple service classes such as DiffServ, ε-probing can eliminate the resource stealing problem, providing an effective solution to support per flow QoS without signaling and without maintaining flow state at core routers.

Keywords: Call Admission Control, DiffServ, QoS, Test-bed.

1 Introduction

The Integrated Services (IntServ) architecture of the IETF provides a mechanism for supporting quality-of-service for real-time flows. Two important components of this architecture are admission control [3], [10] and signaling [5]: the former ensures that sufficient network resources are available for each new flow, and the latter communicates such resource demands to each router along the flow's path. However, the demand for high-speed core routers to process per-flow reservation requests introduces scalability limitations in this architecture.

In contrast, the Differentiated Services (DiffServ) architecture [6], [2] achieves scalability by limiting quality-of-service functionalities to class-based priority

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 49–61, 2002.

mechanisms together with service level agreements. However, without per-flow admission control, such an approach necessarily weakens the service model as compared to IntServ, namely bandwidth or loss guarantees are not assured to individual flows.

A key challenge addressed in recent research is how to simultaneously achieve the scalability of DiffServ and the per-flow QoS assurance of IntServ. Towards this end, several novel architectures and algorithms have been proposed, which require always some specific functionality to be employed at edge and/or core nodes. In probing schemes ([1], [8], [9]), these functionalities are not required: there is no signaling protocol and no special packet processing within core nodes, and still a per-flow QoS is assured. With such a scheme, the endpoints perform admission control by assessing the congestion state of the network, transmitting a sequence of probe packets and measuring the corresponding performance. If the performance (e.g., loss ratio) of the probes is acceptable, the flow is admitted; otherwise it is rejected. More specifically, to establish a real-time flow between two hosts, the sender host transmits a sequence of probes into the network at the desired rate and flow behavior. If the loss ratio of the probes is below a pre-established threshold for the traffic class, then the flow is admitted, and otherwise it is rejected. Scalability is achieved in such a framework by pushing all quality-of-service functionality to end-hosts, indeed removing the need for any signaling or storage of per-flow state. Moreover, [4] found that such an architecture is indeed able to provide a single controlled-load like service as defined in [11].

However, when host-controlled probing schemes are generalized to support multiple service classes, a resource stealing problem, first described in [4], may occur. To illustrate the resource stealing problem, consider the example of a Class-Based Weighted Fair Queuing (CBQ) scheduler, where each of two classes is assigned a weight of 50%. Assume that the offered load is initially $0.8C$ in class 1 and $0.2C$ in class 2, where C is the link capacity. Due to the work conserving nature of the scheduler, class 1 can borrow class 2 resources and utilize up to 80% of the link capacity without loss. If now class 2 probes the link for an additional offered load of $0.3C$, class 2 flows will be admitted and served without loss. However, the service rate of class 1 will decrease to $0.5C$ and 30% of class 1 packets (which belong to already admitted flows) will be dropped. Thus the admission of new flows in class 2 forced class 1 into a situation of QoS violations that can not be detected by the probing flow. Such resource stealing arises from a fundamental observability issue in a multi-class system: the performance isolation property provided by CBQ schedulers also inhibits flows from assessing their performance impact on other classes.

In [7] we proposed ε-probing as a probing scheme designed to eliminate stealing in CBQ schedulers in a minimally invasive way. The goal of ε-probing is to enable inter-class resource sharing to the maximal extent allowed by the system architecture. In ε-probing, a new flow requesting admission in a class transmits a probe in the desired class and, simultaneously, a probe with a small bandwidth ε in all other classes. The motivating design principle is that the impact of the new flow on all classes must be observed, so that the new flow is only admitted if all probes, including the ε-probes, are admitted. Consider again the previous example of the CBQ scheduler. With ε-probing, when class 2 is probed for the additional $0.3C$, class 1 is also probed with ε-probes. The probing in class 2 is successful but the ε-probing in class 1 will not, since class 1 is not allowed to use more bandwidth. Class 2 flows will not be admitted until class 1 releases bandwidth and no resource stealing will occur.

We developed an experimental system with a DiffServ architecture that includes both probing and ε-probing admission control algorithms. The performance of these algorithms is studied through a number of experiments.

The paper is organized as follows. In section 2 we present the experimental system architecture. In sections 3 and 4 we describe two software modules, the traffic generator and the probing module, which were developed as part of the overall experimental system. Section 5 presents the actual experimental set-up used to carry out the experiments. In section 6 we discuss the experimental results. Finally, in section 7 we conclude the paper.

2 Experimental System Architecture

In this section, we describe the experimental system that is designed to evaluate the efficiency of the proposed ε-probing technique, while closely replicating an operational DiffServ network.

The goal in our experimental studies is to observe the behavior of the probing and ε-probing techniques on a congested network. It would be impractical to have the overall traffic demand generated by many different hosts, as it will be the situation in an operational DiffServ network. Instead we have developed a traffic generator software module that, for each Class of Service (CoS), generates traffic at both flow level and packet level. Due to performance reasons, in the actual experimental set-up we use one host for each CoS.

The probing functionality was implemented in a probing software module, which probes on behalf of a set of users. The probing module can be inserted in end-hosts or edge routers. In the actual experimental set-up the probing module is installed in a dedicated PC, called the probing server, which is connected to a local network delimited by two routers, an access router and an edge router. In this configuration, it can be seen as extending the capabilities of current low-cost edge routers to support probing based admission control. The probing module operates in promiscuous mode, by listening to all packets injected into this local network. It accepts flow set-up requests and performs admission control by probing the DiffServ network; it is also responsible for marking the data packets sent by the traffic generators according to requested CoS. The edge router performs packet classification and scheduling, functions that are found in current low-cost routers. The access router is only used for traffic isolation. Thus, the set of two routers plus probing server emulates a DiffServ edge router that includes admission control based on ε-probing.

The interaction between the various network elements is performed by special purpose application layer protocols. The exchange of control information

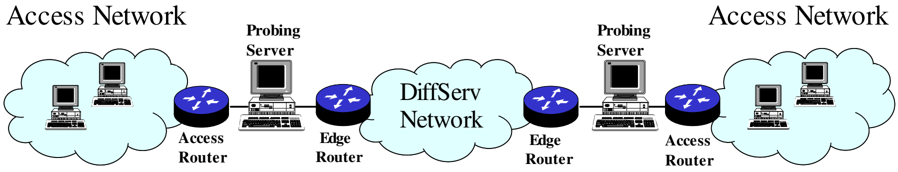


Fig. 1. Experimental system architecture.

between the traffic generator and the probing module is done using TCP. An alternative here could be the use of RSVP. The exchange of control information between probing modules and the data transport is done using UDP.

The message flow is the following (Figure 2). The traffic generator asks for the admission of a new flow by opening a TCP connection with the probing module and sending a REQUEST message. The REQUEST message includes the source and destination IP addresses, the source and destination UDP/TCP ports, the protocol type and the desired class of service. This information is required in order to completely identify the flow at the probing module. Upon receiving the REQUEST message, the ingress probing module initiates the probing process. It sends a PROBE START message, followed by several probe packets, ending with a PROBE STOP message. All these messages are addressed to the destination host and transported over UDP. As mentioned before, there are two types of probe packets: regular probes, sent on the desired class of service, and ε-probes sent on the remaining classes. The egress probing module listens promiscuously to these control messages and probing packets, and counts the number of probes received in each class between PROBE START and PROBE STOP. When it hears the PROBE STOP message it sends a STATISTICS message back to the ingress probing module with this information. If the STATISTICS message is not received within a pre-defined timeout the flow is rejected, and the TCP connection with the traffic generator is closed. Otherwise, the probing module performs an admission control decision based on the counts of probes and ε-probes carried in the STATISTICS message and on the target loss ratio. If the flow is accepted it sends an AUTHORIZE message and closes the TCP connection with the traffic generator; otherwise it sends a REJECT message, also closing the TCP connection. If the flow is accepted the traffic module starts sending data packets (transported over UDP). To signal the end of data transmission, the traffic generator module opens a new TCP connection with the ingress probing module and sends a END SESSION message.

The REQUEST and END SESSION messages have the same format, and are identified by a flag. The AUTHORIZE message corresponds to ”0” and the REJECT message to ”1”, both coded as unsigned int. The probe control messages, PROBE START, PROBE STOP and STATISTICS, include three fields: the first field identifies each message; the second indicates the CoS; the third is used to transport, in the STATISTICS message, the counts of probes in each class. Note that the information exchanged at the application layer is not sufficient to completely identify a flow. The IP addresses and UDP/TCP ports are also required. This option had the purpose of minimizing the overhead.

All sockets used in the communication between probing modules are of type raw sockets. As will be detailed in section 4, this type of sockets allows operation in promiscuous mode and manipulation of the header fields from lower layers.

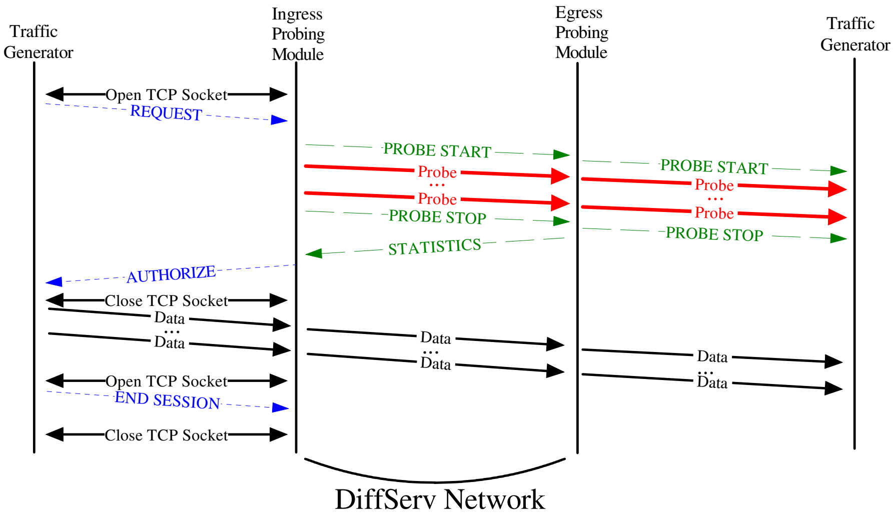


Fig. 2. Message flow between traffic generator and probing modules.

We use the IP TOS byte field to differentiate among classes of service and priorities. It is assumed that control messages injected into the DiffServ network have higher priority. The precedence bits of the TOS byte are used to differentiate between control, probe, ε-probe and data packets. Specifically, we assign 110 to control packets, 010 to probe packets, 100 to ε-probe packets and 000 to data packets. The differentiation between CoS is carried out using the TOS bits of the TOS byte. We leave to the probing module the role of manipulating the TOS byte. All data packets sent by the traffic generator have a TOS byte of zero and are marked according to their class of service at the probing module.

Both the traffic generator and probing modules are developed to run under Microsoft Windows 2000. The software is developed using Microsoft Visual C++, Windows Sockets 2.0 and resorts to multi-thread programming techniques. In our implementation each flow is a thread and, inside each flow's thread, tasks that can be executed concurrently give rise to new threads. The use of Windows Sockets 2.0 and Microsoft SDK make possible the implementation of the promiscuous mode operation at the probing module. Note that the same type of facilities were available for a Unix development.

In the next two sections we will describe with more detail the traffic generator and the probing modules.

3 Traffic Generator Module

The traffic generator module generates the traffic of each CoS at two levels, flow and packet level. It generates new flows according to a Poisson process. The admitted flows have a duration characterized by an exponential distribution. For each flow, the traffic generator creates the corresponding packet stream. Several models are available for the packet arrival process and for the packet length. The arrival process can be Constant Bit Rate (CBR) or ON-OFF with exponential or Pareto ON and OFF durations. CBR sources are only characterized by the packet arrival rate. ON-OFF sources require the specification of the average ON and OFF times and of the packet arrival rate in the ON state. The Pareto distribution requires an additional parameter called shape. The packet length may be fixed, exponential or Pareto.

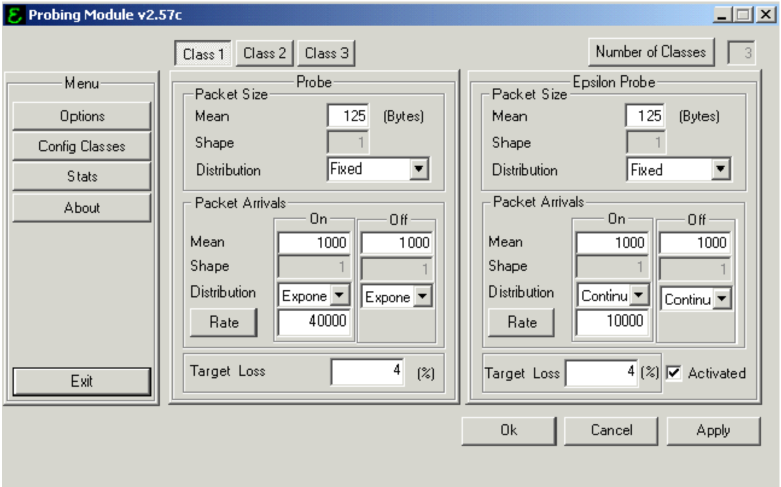
The traffic generator handles two types of sockets: a TCP socket for the exchange of control information with the probing module, and a UDP socket for data transmission. There is also a thread per CoS that schedules the arrival of the next flow and determines its duration. When a flow starts, another thread is created, which is responsible for the generation of packets for that flow, and of the control messages exchanged between the traffic generator and the probing module.

4 Probing Module

The probing module is responsible for handling the probing process and for packet marking. As mentioned above, the probing module listens promiscuously to the packets that are injected into its local network. This mode is implemented using raw sockets, which allows the manipulation of the IP header fields. At the ingress side, the probing module captures the data packets and re-injects them into its local network after changing the TOS and checksum fields of the IP header. Since Microsoft Windows 2000 does not support natively the manipulation of the TOS byte, we developed a patch for this purpose. Besides the raw sockets, the probing module handles a TCP socket for the exchange of control information with the probing module and UDP sockets for the transmission of data packets, probes and ε-probes and probe control messages. There is a thread permanently listening for new flow set-up requests, at a specific port. When the probing module receives a request from the traffic generator, this thread will produce a new one that will handle the flow. To increase the performance of the system, we use asynchronous UDP sockets to prevent the permanent polling of the socket state. The TCP sockets used in the implementation are of blocking type. In this case, the program suspends the execution of other tasks until the socket operation is finished.

The main window allows the configuration of several parameters: the server port for communication with the traffic generator, the gateways towards the access network or the DiffServ network, the probing duration and timeout, and the link capacity. Note that the link capacity is only required for the computation of some parameters (wrong decisions and stolen bandwidth). The timeout indicates the maximum time interval after sending PROBE STOP that the module waits for the STATISTICS message.

Two other windows can be opened from the main one, called probing traffic and statistics, respectively. There is one statistics window for each CoS. The probing traffic window (Figure 3) is where the traffic models and parameters for the generation of probes and ε-probes are configured. It also includes the target loss ratio for probes and ε-probes, and an option for deactivation of ε-probes. The statistics windows includes, for each CoS, statistics such as the number of data packets, probes and ε-probes received and sent, the number of blocked and accepted flows, the number of wrong decisions and the percentage of stolen bandwidth. The window also displays a curve of the evolution of the blocking probability over time. All these parameters are updated in real time. Also, the configuration of the experiment's length and of the warm-up time for statistics collection are performed in this window.



Number of Classes 3

Probe

Packet Size
Mean 125 (Bytes)
Shape 1
Distribution Fixed

Packet Arrivals
On: Mean 1000, Shape 1, Distribution Expone, Rate 40000
Off: Mean 1000, Shape 1, Distribution Expone

Target Loss 4 (%)

Epsilon Probe

Packet Size
Mean 125 (Bytes)
Shape 1
Distribution Fixed

Packet Arrivals
On: Mean 1000, Shape 1, Distribution Continu, Rate 10000
Off: Mean 1000, Shape 1, Distribution Continu

Target Loss 4 (%) ☑ Activated

Fig. 3. Probing traffic window of the probing module.

5 Experimental Set-Up

We perform experiments both with two CoS. All sets of experiments resort to the set-up depicted in Figure 4. Each source host A and B generates traffic in a different CoS. Traffic generator modules are plugged in both source hosts A and B. Hosts A and B are 120 MHz Pentium PCs with 64 Mbytes of RAM. Because of performance reasons two probing servers are used at the ingress side. Probing server A is a 350 MHz Pentium II with 128 Mbytes of RAM, and probing server B is a 733 MHz Pentium III with 256 Mbytes of RAM. The probing server at the egress side is a 933 MHz Pentium III with 256 Mbytes of RAM. The Operating System (OS) of the source and destination hosts A and B is Windows NT 4.0, and the OS of the probing servers is Windows 2000 Professional.

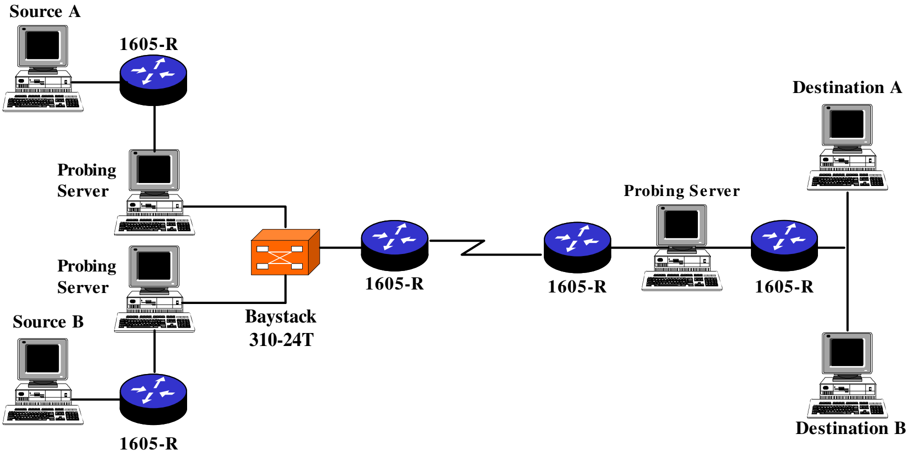


Fig. 4. Experimental Set-Up.

All routers used in the experiments are Cisco 1605 R, running IOS version 12.0(7)T. The ingress and egress edge routers are connected through a serial link, because it offers great flexibility in controlling the link's bandwidth. Our experiments with two CoS resort to Cisco's Custom Queuing. This mechanism works with a maximum of 16 queues, that can be divided in two groups, where one group uses strict priority scheduling and the other uses deficit round-robin scheduling; the latter group has a lower strict priority. In our case, we configure one queue with strict priority (for the control traffic) and two queues with deficit round robin (for the data and probing traffic). Classification at the edge routers is based on the analysis of the precedence and TOS bits and resorts to Cisco's Access Lists. An Ethernet switch (Baystack 310-24T) is used to multiplex the traffic from hosts A and B at the ingress side.

As referred in [4], the probing schemes are able to guarantee a per-flow QoS to controlled load services. The best-effort and the guaranteed services are not considered here, because there will be a different priority for each type of services and a rate limiter will be associated with the guaranteed traffic. Then, our study can be based only on the controlled load services.

6 Experimental Results and Discussion

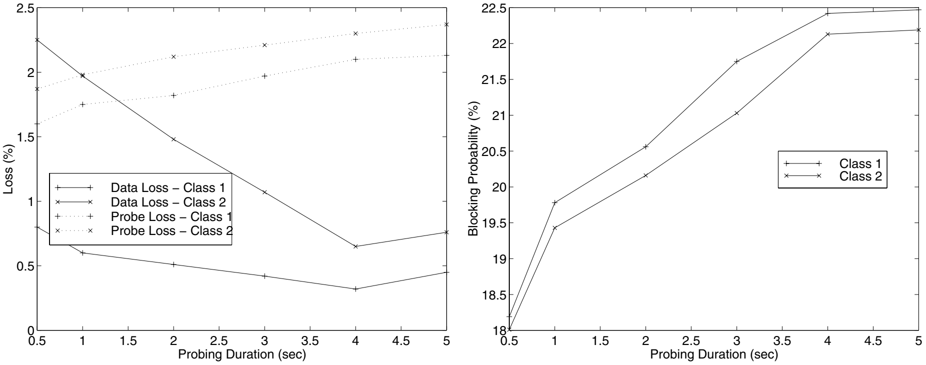
In this section we present and discuss two sets of experiments. The first set considers two CoS and a constant offered load. The second set considers also two CoS but a time-varying offered load.

The traffic sources used in the experiments are always CBR. The arrival and departure rates are adjusted to give blocking probabilities near 0.2 (corresponding to an offered traffic that is approximately 120% of the link capacity). The mean number of active flows in a traffic class is $\rho = \lambda/\mu$, where λ and μ are respectively the mean flow arrival and depart rates. The offered load is given by the mean number of flows (ρ) multiplied by its bandwidth. Unless otherwise specified, the link capacity is 1 Mb/sec, the packet length is 125 bytes, the buffer size of the queues is 24000 bytes and the length of each experiment is 1200 seconds. A warm-up time is used in all experiments, which is at least two times the highest value of the flow's average duration. Note that the probing bandwidth always equals the flow bandwidth. Each experiment described bellow is repeated five times and the results represent the corresponding average values.

6.1 Experiments with two CoS and a Constant Offered Load

This set of experiments addresses two traffic CoS and a constant offered load, i.e., the arrival rate and mean duration of the flows do not vary during the experiments. The goal here is to address the resource stealing problem and analyze the behavior of ε-probing. In all experiments, the target loss ratio of the probes and ε-probes is 5%. The flow's bandwidth is 40 Kb/sec in class 1 and 64 Kb/sec in class 2. The flow's ρ is 4 in class 1 and 11 in class 2. The weight assigned to class 1 is 20% and the weight assigned to class 2 is 80%.

Probing duration. In this experiment the bandwidth of the ε-probes in both classes is 20 Kb/sec and the bandwidth of the probes in each class equals that of the flows requesting admission. Figure 5(a) shows the data and probe loss ratios in each class, as a function of the probing duration. Figure 5(b) shows the corresponding blocking probabilities. The data and probe loss ratios are always below the target, showing that the probing-based admission control operates correctly. The blocking probabilities increase and there is also a slight increase in the probe loss ratio, as the probing duration increases. Except for small probing durations, the data loss is always below the probe loss since not all flows are admitted. For small probing durations, the data loss in class 2 is larger than the corresponding probe loss, which can be explained by lack of accuracy due to insufficient probing duration. The data loss decreases for probing durations between 0.5 and 4 seconds. One might expect that a larger probing time would produce a more accurate estimation of the data loss ratio, i.e., a measured data loss ratio closer to the target (which is 5% in both classes). However, due to the overhead introduced by longer probing times, the effect is the opposite. The same behavior is observed via discrete-event simulation in [4]. For probing durations greater than 4 seconds the data loss ratio increases because the probing traffic gets significant contributing itself to the degradation of the data loss ratio. The loss ratio in class 2 is higher because since class 2 flows have higher bandwidth more probes are generated in the same probing duration.



(a) (b)

Fig. 5. Effect of probing duration on (a) data and probe loss, and (b) blocking probability.

Mismatch between offered load and CBQ weight. In this experiment we introduce a mismatch between the offered load and the CBQ weight of class 1. The weight is kept as before at 20%; the offered load is increased from 20% to approximately 50% of the link capacity (by increasing ρ from 4 to 11). In class 2 we keep everything as before. In Figure 6 we show the data and probe loss ratios versus the bandwidth of ε-probes. With no ε-probing (a null ε-probe bandwidth) the data loss in class 2 is almost 8%, which is larger than the threshold. This behavior is maintained for ε-probe bandwidths bellow 4 Kb/sec, and can be attributed to resource stealing. In fact, whenever class 2 goes into underload, class 1 flows will try to use some of the fair-share bandwidth of class 2 with success. Class 1 flows will then experience resource stealing because in this situation, and since probing is only in the requested class, new requests for class 2 flows will be accepted (at the cost of stealing bandwidth to already accepted class 1 flows). Figure 6 also shows that the probing loss in class 2 increases with the ε-probe bandwidth. This increase is responsible for blocking more class 2 flows when class 1 is using some of the fair-share bandwidth of class 2, which reduces the bandwidth stealing in class 1.

6.2 Experiment with two CoS and a Time-Varying Offered Load

In this experiment we consider a time-varying offered load. The motivation here is to increase the potential for resource stealing, in order to study the effectiveness

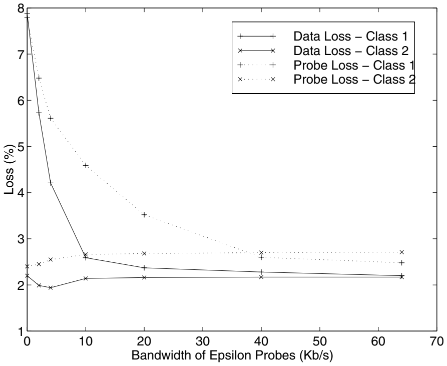
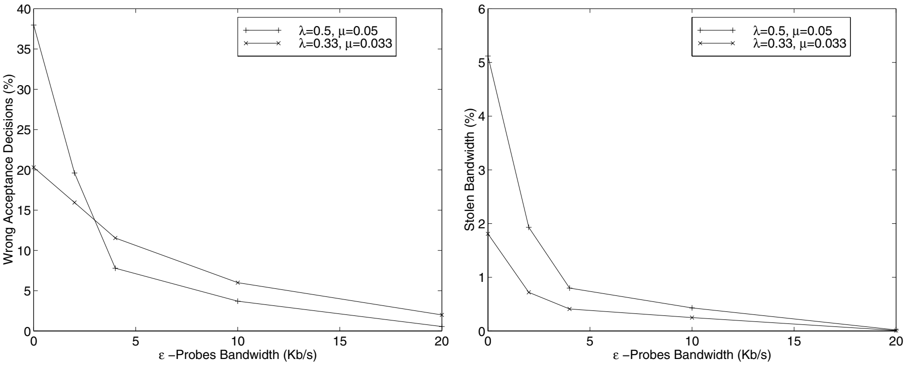


Fig. 6. Effect of mismatch between offered load and CBQ weights on data and probe loss.

of the ε-probing scheme. Specifically, we increase the traffic intensity of class 1 during the experiment at a specific time instant, coinciding with the start of data collection for the purpose of statistics computation. Before this perturbation, the offered load is 20% of the link capacity in class 1 and 80% in class 2. Since each class is assigned a weight of 50%, class 1 will be underloaded and class 2 will be overloaded. The experiment consists in increasing the offered load of class 1, to force the bandwidth stealing of already accepted flows from class 2. The offered load in class 2 corresponds to 64 Kb/s of flow bandwidth and a mean number of flows ρ of 11. Before the perturbation, the offered load in class 1 corresponds to 64 Kb/s of flow bandwidth and a mean number of flows ρ of 4.

The probing duration is kept constant at 2 seconds. The model of the traffic source is CBR in all cases. The target loss ratio of probes and ε-probes is 5%. In the actual experiment, the increase in traffic intensity of class 1 is implemented by two traffic generators. Both generators have a constant offered load, but the second one is only activated later in the experiment. We consider two cases for the perturbation: the second generator has (i) an arrival rate of $0.5sec^{-1}$ and ρ of 10; (ii) an arrival rate of $0.33sec^{-1}$ and also a ρ of 10; the flow bandwidth is kept at 64 Kb/sec in both cases. The goal is to keep the traffic intensity approximately constant while increasing the arrival rate. Given that we want to analyze the transient behavior of the system, i.e., when a perturbation arises, the length of the experiment was constrained to 200 sec (from the start of the second generator), to avoid averaging out the stealing effects.

To analyze the results of the experiment we use two performance metrics: the percentage of wrong decisions and the percentage of stolen bandwidth. The former is the percentage of flows that are accepted when the bandwidth of all admitted flows is higher than the link capacity. The latter is the percentage of bandwidth that is stolen by the admission of new flows when this admission is a wrong decision. The computation of these metrics is done as follows: whenever there is a positive admission decision, we calculate the bandwidth occupied by all admitted flows, based on the number of flows and on the flow's bandwidth. If this bandwidth is larger than the link capacity (including the tolerance given by the loss target), the decision is computed as a wrong decision. In this case, the difference between the bandwidth of the admitted flows and the link capacity is the stolen bandwidth.



(a) (b)

Fig. 7. Effect of time-varying offered load on (a) wrong decisions and (b) stolen bandwidth.

Figure 7(a) and (b) show that without ε-probing the percentage of wrong decisions and of stolen bandwidth is very high (wrong decisions are 38% with the first perturbation and 20% with the second one; stolen bandwidth is more than 5% in the first perturbation and almost 2% in the second one). This is due to resource stealing, when class 1 recovers its bandwidth after the system's perturbation. Both metrics decrease rapidly with the ε-probe bandwidth: with only 2 Kb/sec the stolen bandwidth values decrease almost to one half, and with 10 Kb/sec (less than 1/6 of the bandwidth of admitted flows) the stealing is almost insignificant. A comparison of the two curves in each figure shows that a larger arrival rate provokes more stealing. Thus, the results of this experiment where resource stealing is intentionally aggravated, clearly show that ε-probing is able to eliminate this problem.

7 Conclusions

Placing admission control functions at the network's endpoints has been proposed as a mechanism for achieving per-flow quality of service in a scalable way. In this paper we have described an experimental system with a DiffServ architecture that includes both probing and ε-probing admission control algorithms. The ε-probing technique was introduced to overcome the so-called resource stealing problem that impairs multi-class systems based on simple probing. A number of experiments was carried out to study the performance of these admission control algorithms. The results clearly show that the probing schemes are able to accurately perform admission control while achieving high utilization. Moreover, they also show that in multi-class environments such as DiffServ, ε-probing can eliminate the resource stealing problem. For example, it was shown that the resource stealing problem can be virtually eliminated by using ε-probes with a bandwidth higher than 1/6 of the flows' bandwidth. Thus, the ε-probing scheme is able to provide an effective solution to support per- flow QoS without signaling and without maintaining any flow state at core routers.

References

1. G. Bianchi et al. Throughput analysis of end-to-end measurement-based admission control in ip. In *Proceedings of IEEE INFOCOM 2000, Tel Aviv, Israel*, March 2000.
2. K. Nichols et al. *Two-bit differentiated services architecture for the Internet.* Internet RFC 2638, 1999.
3. L. Breslau et al. Comments on the performance of measurement-based admission control algorithms. In *Proceedings of IEEE INFOCOM 2000, Tel Aviv, Israel*, March 2000.
4. L. Breslau et al. Endpoint admission control: Architectural issues and performance. In *Proceedings of ACM SIGCOMM 2000, Stockholm, Sweden*, August 2000.
5. L. Zhang et al. Rsvp: A new resource reservation protocol. In *IEEE Network*, volume 7, pages 8–18, September 1993.
6. S. Blake et al. *An architecture for differentiated services.* Internet RFC 2475, 1998.
7. S. Sargento et al. Resource stealing in endpoint controlled multi-class networks. In *Proceedings of International Workshop on Digital Communications (Invited Paper), Taormina, Italy*, September 2001.
8. V. Elek et al. Admission control based on end-to-end measurements. In *Proceedings of IEEE INFOCOM 2000, Tel Aviv, Israel*, March 2000.
9. R. Gibbens and F. Kelly. Distributed connection acceptance control for a connectionless network. In *Proceedings of ITC '99, Edinburgh, UK*, June 1999.
10. E. Knightly and N. Shroff. Admission control for statistical qos: Theory and practice. In *IEEE Network*, volume 13, pages 20–29, March 1999.
11. J. Wroclawski. *Specification of the controlled-load network element service.* Internet RFC 2211, 1997.

High Performance DiffServ Mechanism for Routers and Switches: Packet Arrival Rate Based Queue Management for Class Based Scheduling

Bartek Wydrowski and Moshe Zukerman

ARC Special Research Centre for Ultra-Broadband Information Networks,

EEE Department, The University of Melbourne,

Parkville, Vic. 3010, Australia

{ b.wydrowski, m.zukerman }@ee.mu.oz.au

Abstract. This paper introduces a technique for applying packet arrival rate based queue management to class based scheduling algorithms. This enables a DiffServ architecture with very low packet latency, loss, and high link utilisation. Simulation results demonstrate that the proposed technique outperforms the current weighted random early drop (WRED) and weighted fair queue (WFQ) architecture.

1 Introduction

At the core of the Internet's Differentiated Services (DiffServ) architecture are the packet scheduling and queue management algorithms in routers or switches. Today's premier DiffServ architecture consists of a weighted fair queue (WFQ) with weighted random early drop (WRED) queue management. However, literature has shown that performance of packet arrival rate based congestion control, such as REM [5] or GREEN [8], significantly outperforms packet backlog based techniques such as drop-tail or RED. In this paper we form a basis for a high performance DiffServ architecture by applying rate-based queue management to packet scheduling algorithms. In the following subsections we give an overview of the area and show why rate-based control with packet scheduling is desirable.

1.1 Congestion Control Overview

Asides from the physical capacity of the network, the key design component that determines the quality of service of packet networks is load control. Load control determines how many packets are allowed onto each link of the network, who gets to send them and when. This controls the bandwidth, latency and jitter experienced by users.

There are a number of load control mechanisms, characterised by the amount of connection state information stored in the network. The range goes from connection admission control schemes, such as RSVP, through to stateless congestion control such as TCP, which is a subset of a more general macro-economic like system [7]. Diffserv occupies a middle ground, where individual connections are controlled on a connectionless/stateless basis from the perspective of the network, but aggregates of

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 62-73, 2002.

flows, i.e. classes, receive pre-configured treatment at links, which require per-class information. This results in a good compromise between system complexity and control of performance.

Congestion control on an Internet with *Diffserv* is performed by two independent and concurrent mechanisms: (1) a closed-loop control mechanism controls the transmission of packets onto the end-to-end source to destination paths, and (2) open-loop packet scheduling algorithms, enforce statically pre-configured prioritisation and allocation of bandwidth at each link.

The closed-loop congestion control system consists of source algorithms controlled by link algorithms. The source algorithm is any protocol which transmits onto the Internet (e.g. TCP, UDP, RTP etc.) and is not necessarily responsive to congestion. The link congestion control algorithm is sometimes called queue management or active queue management (AQM). Examples of AQM algorithms include drop-tail, Random Early Drop (RED) [3] (and variants: WRED [1], GRED [14] etc.), Random Exponential Marking (REM) [5], Blue [4] and GREEN [8]. The AQM algorithm signals congestion to the source by packet marking, namely, explicit congestion notification (ECN) or packet dropping.

The open-loop scheduling algorithms determine which packet to send next [6]. They decide the order of packet transmission based on the order of packet arrival and the packet priority class. DiffServ uses a 3 bit code in each packet to identify the class. Scheduling of the packet controls the order of transmission as well as the relative bandwidth allocation to each class. Examples of scheduling disciplines include First In First Out (FIFO), Round Robin (RR), Priority Scheduling (PS) and Weighted Fair Queueing (WFQ).

1.2 Need for Scheduling: Classless vs. Class Based Differential Service

A number of papers [5] [7] have proposed an architecture for differentiated services without explicit packet classes. Instead, sources differentiate their demand for bandwidth by utility functions, $x = \mathrm{U}(p)$, which determine the source's transmission rate x based on the current network price p. The price p, is determined by the end-to-end congestion level, and is communicated to sources from the AQM algorithm by packet marking or dropping. In fact, the network functions as a macro-economic system, where links sell their bandwidth and sources purchase it, based on their utility function. Sources which require more bandwidth than others, simply send more, suffering a higher price p. In such a system, scheduling algorithms are redundant because the allocation of bandwidth is determined solely by the macro-economic process. It has been shown that such a system maximises the aggregate of the utilities of all the sources [7].

If maximising the aggregate utility of the system is the only criteria, this system is sufficient. However, no guarantees can be made about the amount of bandwidth actually allocated to each source, because the current 'market' of all sources on the network determines this. A real network will consist of a subset of sources which require a minimum rate guarantee, and a subset which are satisfied by their 'market-share'. Since the network administrator is not aware of all of the utility functions of all the flows traversing the network, it is not possible to configure the utility functions of sources to guarantee their minimum rates in a competitive environment.

Many real applications require minimum rate guarantees. For example, an office with a set of voice-over IP telephones, an interactive online game, or video-conferencing, all require a guaranteed amount of bandwidth from the network at any time, regardless of the background traffic. If a subset of sources needs a guaranteed minimum rate from a link, the macro-economic system is not sufficient. A flow isolation mechanism, which removes the flows needing guarantees out of the competitive macro-economic environment that contains other flows with unknown utility functions, is essential to guarantee minimum rates. DiffServ with packet marking and link class-based scheduling does this by guaranteeing minimum capacity to flow subsets.

In practice, IP router manufactures have recognised this need for scheduling algorithms. However, the existing architecture for congestion control in a class-based environment remains crude. Until now, the closed-loop congestion control, or queue management, within the scheduling mechanism has been based on backlog measuring techniques such as drop-tail, RED or WRED. In this paper, a technique for implementing high performance rate based congestion control in a scheduler such as WFQ is introduced.

1.3 Need for Rate Based Control: Rate vs. Backlog Based Congestion Control

Broadly, there are two paradigms of congestion control algorithms, characterised by the way they observe congestion. Backlog based (BB) control, like droptail, RED and WRED, measures the number of packets in the buffer to determine the severity of congestion. Arrival rate based (RB) control schemes, such as REM or GREEN, measure the packet arrival rate.

In general, AQM algorithms signal congestion to the source algorithms by varying the rate of packet dropping or ECN marking, P. For BB control, P is a function of the backlog size $b(t)$ at time t, $P(t)=f(b(t))$; where $f(x)$ is a positive and increasing function for $x > 0$ and $f(0)=0$. For RB control, P is typically driven by an integration process, which sums the excess demand, such as $P(t+1) = P(t) + \Delta P\times(x(t) - u\times c(t))$, where $x(t)$ and $c(t)$ are the arrival and service rates at time t respectively, ΔP is the gain of the control which affects the stability and convergence, and u controls the target utilisation. Although BB congestion control is simpler to implement, it has some inherent limitations not present in RB control.

The backlog (queuing) process $b(t+1)=[b(t) + x(t) - c(t)]^+$, cannot observe long term arrival rates $x(t) < c(t)$ as if $x(t) < c(t)$ for a sufficient period of time, then b(t) reaches zero. Once $b(t)=0$, and if $x(t)$ continues to be less than $c(t)$ and b(t) remains zero, we can say nothing about how close $x(t)$ is to $c(t)$ by observing the state of $b(t)$. Therefore, by observing $b(t)$ the sources cannot be provided with feedback about the level of $x(t)$ to control their transmission rate, as $b(t)$ stays at zero and provides no information about $x(t)$. Given that a positive feedback signal P is required to control the source at some steady rate $x(t)$ where $x(t) < c(t)$, and $P(t)=f(b(t))$, the backlog must be positive, $b(t) > 0$, for P to be positive. This shows how BB control posits the existence of backlog and backlog is necessary for the control process itself.

Backlog is undesirable because it creates packet latency and delay jitter. Furthermore, delay in the congestion control system loop pushes the network towards instability, increasing the likelihood of buffer overflow and under-utilisation. Of course, some backlog is necessary to achieve a desired utilisation of a link with a non-deterministic arrival process, however this is at worst equal to, but typically far less than, the backlog created by BB control such as drop-tail or RED [8].

Unlike the BB schemes, the RB control mechanism can observe *x(t)* directly. In a steady state situation, where the input process is stationary, the amount of backlog kept can therefore be only the minimum required to achieve the desired utilisation. It is not the intention of this paper to give a thorough performance comparison of different congestion control strategies, only to indicate some of the reasons why it is desirable to have a RB control strategy. For more background, the reader is referred to [5] [8].

Now that we have presented the need for (1) class based scheduling algorithms and (2) RB control, the algorithm which combines the two is presented in Section 2 and its performance evaluation is presented in Section 3.

2 Algorithm Background

RB AQM operates in symbiosis with a scheduler. Our proposed design of RB AQM applies to a work conserving WFQ like scheduler. A work conserving scheduler is never idle if there are any packets in any queue. A WFQ like scheduler, such as RR and many variants of WFQ, allocates a portion of service time to each queue during an interval of operation. The scheduler is interfaced to by the enqueue and dequeue functions, which accept and provide the next packet for queuing or transmission respectively.

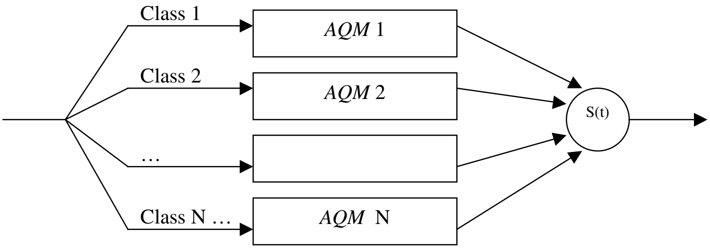


Fig. 1. RB AQM architecture in a class based scheduler

As shown in Fig. 1, each queue in the scheduler is managed by a separate instance of an AQM algorithm. The AQM algorithm decides which packets to drop or ECN mark. Packet marking/dropping gives the source algorithm a feedback signal which controls its transmission rate and avoids queue overflow or excessive backlog. Traditionally, this would be performed by BB control, such as drop-tail or RED queue. RB control directly replaces these algorithms. In general, RB AQM is any process which determines the packet marking/dropping rate, *P(t)*, from at least the packet arrival rate *x(t)* and capacity *c(t)*. Typically, the process for *P(t)* is an integrator of excess demand [10], $P(t+1) = P(t) + \Delta P\times(x(t) - u\times c(t))$, however, other functions are possible, motivated by better convergence or stability (eg: REM, GREEN).

$$P_i(t+1) = AQM(c_i(t), x_i(t), P_i(t), \ldots) \qquad 1 \geq P_i(t) \geq 0 \tag{1}$$

The distinctive issue, faced by RB AQM in a class-bases scheduler, is that the capacity available to each class *i*, denoted c_i, and the packet arrival rate for that class, denoted x_i, need to be known. In work conserving scheduler, such as WFQ, where unused capacity in one class is redistributed to other classes, the capacity available to each class is time-varying and depends on, and affects, the traffic in other classes. This paper enables RB AQM by presenting a technique for calculating and controlling c_i, the capacity allocated to each class. Class Dimensioning, or controlling the number of users per class is beyond the scope of this paper.

A basic algorithm is introduced in Subsection 2.1 which results in a functional work conserving RB system, where each class is guaranteed its minimum share, M_i. However, the capacity above the minimum is not distributed with any notion of fairness. Instead, the classes with the most aggressive traffic win the slack capacity. In Subsection 2.2, we present a notion of proportional fairness, and a mechanism to enforce it.

2.1 Basic Algorithm

2.1.1 Capacity Estimation

Consider a stream of packets scheduled by a work-conserving WFQ scheduler, of *N* classes. Let *B* be the vector representing the sizes (bits) of the *H* packets that have been served most recently. The order of the elements of vector *B* are in reverse order to their service completion times. In other words, B_0 is the size of the most recently served packet, B_1 is the size of the previous packet and so on. Finally, B_H is the size of the oldest packet packet in *B*. Similarly, we define the vector *C*, of *H* elements, such that C_j is the class (Cj ∈ {1, 2, 3, … N}) of the packet represented by B_j, j = 1, 2, 3, … *H*.

Let *S(t)* be the physical capacity of the link at time *t*. When *S(t)* is time varying, such as with Ethernet, DSL, or radio, it can be estimated from the last packet's transmission time. The scheduling algorithm, such as WFQ, may guarantee minimum rates to each class. Let *W* be a vector whose element W_i corresponds to the share of capacity that each class *i* is guaranteed. For a WFQ scheduler, W_i corresponds to the service quantum for class *i*.

In a work conserving scheduler, the actual capacity available to a class depends on the traffic in other classes as well as on the minimum rate allocation *W*. Without apriori knowledge of the traffic, the future capacity available to a class, can only be estimated from the previous capacity. Let the identity function *I(j,i)* be:

$$I(j,i) = \begin{cases} 1 & \mathit{if}\ C_j = i \\ 0 & \mathit{otherwise.} \end{cases} \tag{2.1}$$

The estimate class capacity, $S_i(t)$, is calculated from the portion of server time allocated to class *i* by the scheduling mechanism in the past *H* packets:

$$S_i(t) = \frac{\sum_{j=0}^{H} B_j(t) \cdot I(j,i)}{\sum_{j=0}^{H} B_j} S(t) \qquad \text{where} \quad i < N. \tag{2.2}$$

Note reduced complexity techniques such as exponential averaging could be employed to compute (2.2).

2.1.2 Capacity Allocation

The minimum service rate guaranteed by the WFQ scheduling mechanism, M_i, is given by:

$$M_i(t) = \frac{W_i}{\sum_{j=1}^{N} W_j} S(t) \cdot \tag{3}$$

The capacity allocated to each class is therefore also bounded by the minimum rate enforced by the WFQ scheduling policy. The capacity allocated to class i, denoted $c_i(t)$, is:

$$c_i(t) = Max(M_i(t), S_i(t)). \tag{4}$$

Notice that $c_i(t)$ is the capacity allocated to class i, not the capacity actually consumed by class i. The capacity not consumed by the class to which it is allocated, may be used by other classes. If for example, no class i packets arrive, $s_i(t)$ will be 0, and $c_i(t)=M_i(t)$. Although in this case no capacity is consumed by class i, if a burst of class i packets were to arrive, $M_i(t)$ capacity is guaranteed. Note (4) is evaluated at each update of the AQM process (1), which at the maximum rate, is at every enqueue event.

2.2 Extended Fair Share Algorithm

The algorithm in 2.1 is extended here to enforce a notion of proportional fairness. The fair allocation enforcement applies only to bottlenecked classes, where $x_i(t) \geq c_i(t)$. Classes which are not bottlenecked at the link, $x_i(t) < c_i(t)$, need no enforcement of fairness, since their rate is below their fair capacity and their bandwidth demand is satisfied. We define a fair allocation of capacity to a bottlenecked class i, $F_i(t)$, as:

$$F_i(t) = \frac{W_i}{\sum_{j=\text{all bottlenecked classes}} W_j} (S(t) - \sum_{j=\text{all non-bottlenecked classes}} x_j). \tag{5}$$

In the extended algorithm, the capacity of non-bottlenecked classes is given by (4), and for bottlenecked classes, the capacity is given be (5). Notice that the sum of *$c_i(t)$* for non-bottlenecked by (4) and *$F_i(t)$* by (5) may be more than *S(t)*. However, the non-bottlenecked classes do not utilise their allocated capacity *$c_i(t)$*, and the aggregate arrival rate is controlled below the capacity *S(t)*.

3 Implementation and Transient Performance Evaluation

3.1 Implementation

For class *i*, RB control was implemented in a WFQ scheduler with a variation of GREEN as the AQM algorithm, as follows:

$$P_i(t) = P_i(t) + \Delta P_i(t) \cdot U(x_i(t) - u_i \cdot c_i(t)) . \tag{6.1}$$

where

$$U(x) = \begin{cases} +1 & x \geq 0 \\ -1 & x < 0 \end{cases} \tag{6.2}$$

and

$$\Delta P_i(t) = \max(abs(x_i(t) - u_i \cdot c_i(t)), k) . \tag{6.3}$$

where u_i controls the target utilisation and hence also the level of queuing, and k is a constant which limits the minimum adjustment to *$P_i(t)$*, to improve convergence. The values of *$P_i(t)$*, *$x_i(t)$* and *$c_i(t)$* are updated with every class *i* packet arrival. The pseudo-code for the WFQ scheduling algorithm used is:

```
pkt*  f  d    ()

     ( RU )

        fo       to
                f (     [      tpkt         [  )

                        [      [          [      tpkt     ()
                         t    (     [   d      )


        fo       to
                f ( [          )
                        [      [       [



 f       (pkt *p  k t)

             [p  k t              (p  k t)
```

Fig. 2. Low jitter WFQ scheduler

This particular WFQ variant minimizes the jitter of higher priority classes, lower class number. The *wfq.deque* function is invoked when the link is ready to transmit the next packet and the *wfq.enque* function is invoked when a packet is received for transmission onto the link. A packet queued in a higher priority class will always be served next, so long as the class's work quantum, *W*, has not been exceeded. Note that the function *class[I].nextpkt.size* returns the size [bits] of the next packet in class *I*, or infinity if there are no packets left in the class. The constant *MaxS* controls the maximum burst size allowable to be transmitted in a class that has been previously idle.

3.2 Performance Evaluation

The system was simulated using Network Simulator 2 [9]. Three scenarios simulated are presented in this paper. All scenarios used the same network topology, as depicted in Fig. 3. For Scenarios 1 and 2, the Diffserv managed link *X* has a 1 Mbps capacity and it is 2Mbps in Scenario 3. Multiple TCP or UDP sessions are aggregated to form the traffic of each of the four classes presented to the link. All data packets are 1000 bytes. We will now describe each simulation scenario and the results.

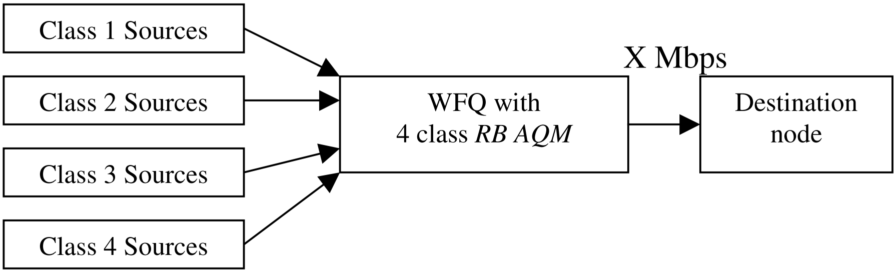


Fig. 3. Overview of Simulation Topology

Scenario 1A and 1B: TCP Traffic

The traffic of this scenario consists only of TCP sources. Scenario 1A uses RB and WFQ with the fairness enhancement (5). Scenario 1B uses WRED and WFQ. The flow rates of traffic in each class and the total number of packets backlogged for all classed was measured. The parameters for this scenario are listed in Table 1.

Table 1. Simulation Parameters for Scenario 1

Class	u_i *Utilisation*	W_i	Sources	Start (sec)	Stop (sec)
1	0.93	8001	8 TCP 40ms RTT	0	100
2	0.93	4001	8 TCP 40ms RTT	20	140
3	0.93	2001	16 TCP 40ms RTT	40	180
4	0.93	1001	16 TCP 40ms RTT	60	220

The WRED implementation uses a weighted average of backlog, denoted $B_w(t)$, to determine the packet marking/dropping probability. The marking probability is related linearly to $B_w(t)$, by $P(t) = \alpha B_w(t)$, where α is the reciprocal of the maximum queue size q. In Scenario 1B q equals 10.

Fig. 4 confirms that a fair allocation of capacity is achieved with the RB and WFQ, as the magnitude of the flow rate from each class is proportional to its minimum rate *W* when the traffic from different classes is switched on and off.

Figures 5 and 7 show the backlog of the RB and BB (WRED) system, with the thick black line being the average backlog measured over 300 packets. The figures illustrate the poorer queuing performance of WRED and WFQ compared to RB and WFQ congestion control. In the interval 50s to 100s, when all classes are active, note how backlog increases with increasing traffic load. This illustrates the previous analysis, that with BB control where *P(t)=f(b(t)),* backlog is necessitated by the control system. With increased traffic load, the feedback signal *P(t)* must also increase to control the sources, and since *P(t)* is coupled with backlog, the backlog must also increase. Compare this with RB congestion control in Fig. 5, where the backlog varies about 0 regardless of the traffic.

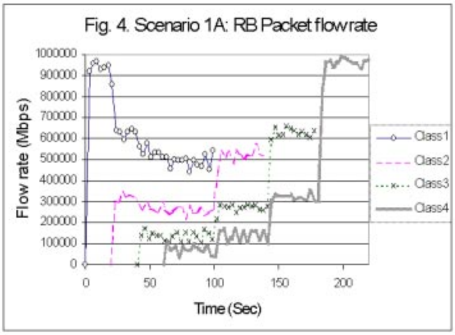


Fig. 4. Scenario 1A: RB Packet flowrate

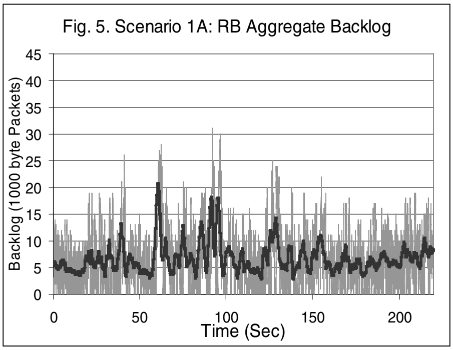


Fig. 5. Scenario 1A: RB Aggregate Backlog

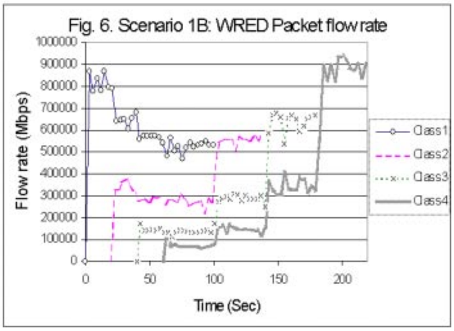


Fig. 6. Scenario 1B: WRED Packet flowrate

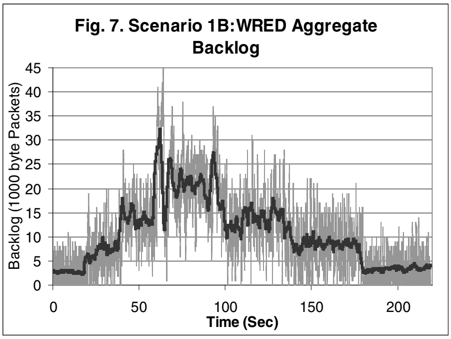


Fig. 7. Scenario 1B:WRED Aggregate Backlog

Scenario 2: TCP and UDP Traffic

This traffic scenario consists of both UDP and TCP sources. Classes 1 and 4 are UDP constant bit rate sources transmitting at 0.8 Mbps and 0.05 Mbps respectively. UDP sources ignore congestion notification. Classes 2 and 3 are comprised of TCP sources. For the complete parameters refer to Table 2.

Fig. 8 shows that RB control allocates bandwidth fairly, despite the presence of an unfriendly, non-congestion-controlled UDP sources. Notice that at 50sec, when Class 2 traffic is switched on, the UDP traffic in Class 1 is throttled down to its fair share by an increased packet dropping rate. At this point Class 1 becomes a bottlenecked class.

In this way, the TCP sources can attain their fair share despite the aggressive UDP source.

Fig. 8. Scenario 2: Packet flow rate in all classes

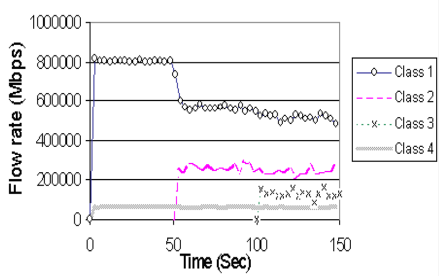
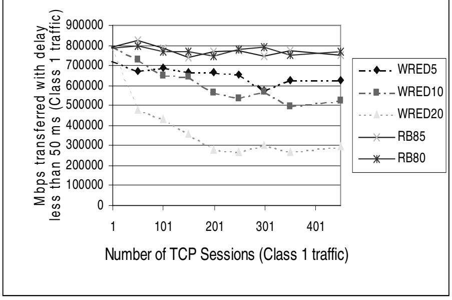


Fig. 9. Scenario 3: WRED and RB Delay Performance



Scenario 3: Real-Time Traffic

In this scenario, it is demonstrated how a RB Diffserv architecture outperforms BB control for real-time traffic. Two classes are used to simulate the interaction of data traffic and real-time traffic. Class 2 contains TCP/FTP data traffic, and is insensitive to delay. Class 1 is the real-time traffic, with a hard maximum queuing delay requirement of 50 ms. The traffic in Class 1, the real-time traffic, consists of saturated TCP transfers, with the number of sessions increasing linearly from 1 to 450. A number of trails were simulated, using WRED with queue size value q set to 5 (WRED5),10 (WRED10) and 20 (WRED20) packets, and using RB control with parameter u_1 set to 0.8 (RB80) and 0.85 (RB85). The Diffserv link capacity is 2Mbps, with 1Mbps assigned to Class 1 and 1Mbps assigned to Class 2.

Table 2. Simulation Parameters for Scenario 2

Clas s	u_i *Utilisation*	W_i	Sources	Start (sec)	Stop (sec)
1	0.93	8001	1 UDP 20ms RTT 0.8 Mbps	0	150
2	0.93	4001	16 TCP 20ms RTT	50	150
3	0.93	2001	16 TCP 20ms RTT	100	150
4	0.93	1001	1 UDP 20ms RTT 0.05 Mbps	0	150

Table 3. Simulation Parameters for Scenario 3

Class	u_i *Utilisation*	W_i	Sources	Start (sec)	Stop (sec)
1	0.8, 0.85	2001	50-450 TCP 40ms RTT	0	450
2	0.95	2001	8 TCP 40ms RTT	0	450

TCP is used to approximate a real-time adaptive multi-rate source [11] [12] [13]. Audio and video protocols are typically based on UDP, RTP and RTCP. Recent real-time multimedia protocols respond to loss by adjusting their rate, and are thus in principle similar to TCP [11] [13]. Although their transient behaviour, and amount of response to loss is different than TCP, any real-time protocol that seeks to take advantage of available capacity on a best effort network, must in principle be congestion controlled. Unless the real-time source increases its rate when there is available capacity, and decreases it when capacity decreases, the quality of transmission is suboptimal. Many existing CODECS are designed for varying channel conditions, such as a best effort network. For instance, the G.723.1 Audio speech codec adjusts its output rate, and adapts to the available bandwidth. Similarly, MPEG-4 includes extensive support for multi-layered, multi-rate video. The RTP communicates the amount of packets lost, which allows the sender to adapt its rate to the channel. At a bottleneck link, adaptive multimedia sources are like saturated sources, such as an FTP transfer, as the source always has more video or audio information that it could possibly send to improve quality.

In the simulation we measure the amount of packets, in Mbps, which are delivered with less than 50ms queuing delay in the Diffserv queue. Packets served late, >50ms, no longer contain useful information to a real-time application and do not contribute to the Mbps. Since real-time sources do not retransmit packets, the TCP packet retransmissions are considered as new packets in the simulation. The results, in Fig. 9, show how for a variety of settings, and traffic loads, RB control effectively delivers more useful data.

As discussed previously, the problem with BB schemes such as WRED, is that the backlog must be positive for source rate to be controlled. In this trial, the maximum queue size for WRED was reduced from 20 to 10 and then to 5. Reducing the maximum queue size gave diminishing returns since the utilisation was significantly lowered. On the other hand, increasing the queue size resulted in a higher average backlog, which delayed more traffic beyond the 50ms requirement. Also, as evident in Fig. 9, unlike RB control, the optimal setting of parameters for WRED varied widely with the traffic load. RB control was able to deliver more data in the delay specification, since it was able to control the arrival rate to some specified fraction below the service capacity, leaving spare capacity for the bursts in the traffic.

3.3 UDP: Throw Away – No Delay

In result in this section we focused on the possible disruptive effect of UDP traffic on TCP traffic, or the interaction between TCP traffic in different classes. An important issue is the performance of non-congestion controlled UDP traffic. UDP is typically used for real-time services with an upper bound delay requirement. If such traffic receives enough capacity, both BB and RB schemes function identically. However, when the amount of non-congestion controlled UDP traffic exceeds the capacity, BB schemes, such as WRED will increase backlog and delay, whereas RB control will prevent excessive delay by increasing the dropping rate. This means, that is instead of being excessively delayed, packets are discarded. Therefore in a congestion situation, the portion of packets which are transmitted, still meet the delay requirements. The portion which are discarded would likely not have been able to be served within the delay requirement. With WRED, in a congestion situation, the delay performance of all packets suffers.

4 Conclusion

We have presented a technique for applying rate based active queue management to a class based scheduling algorithm. The method presented is scalable, and low in computational complexity. It forms a solid architecture for DiffServ implementation in routers and switches and has been shown to outperform the current WRED with WFQ architecture. Furthermore, this work will enable the wide body of research into rate based congestion control schemes to be applied to improving the performance of DiffServ.

References

1. Cisco Systems Document, “Class-Based Weighted Fair Queueing”
2. Cisco Systems Document, “Low Latency Queueing”, http://www.cisco.com/warp/public/732/Tech/qos/techdoc/diffserv.shtml
3. S. Floyd and V. Jacobson, “Random early detection gateways for congestion avoidance” IEEE/ACM Transactions on Networking, 1(4):397--413, August 1993.
4. Wu-chang Feng, Dilip D Kandlur, Debanjan Saham Kang G.Shin, “Blue: A new class of active queue management”. Department of EECS University of Michigan
5. S. H. Low and D. E. Lapsley, “Optimization Flow Control, I: Basic Algorithm and Convergence”, *IEEE/ACM Transactions on Networking*, vol 7 part 6 pp861-875, Dec. 1999.
6. Internet Engineering Task Force IETF, “Recommendations on Queue Management and Congestion Avoidance in the Internet”, RFC 2309.
7. F. P. Kelly, A.K. Maulloo and D.K.H, “Rate control in communication networks: shadow prices, proportional fairness and stability”, Tan (Statistical Laboratory, University of Cambridge), *Journal of the Operational Research Society*, vol. 49, pp 237-252. 1998
8. B. Wydrowski and M. Zukerman, “GREEN: An Active Queue Management Algorithm”, 2001, (submitted for publication, available: http://www.ee.mu.oz.au/pgrad/bpw).
9. The Network Simulator - ns-2 homepage: http://www.isi.edu/nsnam/ns/
10. F. Paganini, J. C. Doyle and S. H. Low, “Scalable Laws for Stable Network Congestion Control”, submitted to CDC01. March 2, 2001.
11. J. Padhye, J. Kurose, D. Towsley, and R. Koodli, "A model based TCP-friendly rate control protocol," in Proc. International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV), Basking Ridge, NJ, June 1999.
12. I. Busse, B. Deffner, and H. Schulzrinne, "Dynamic QoS control of multimedia applications based on RTP," Computer Communications, Jan. 1996.
13. R. Rejaie, D. Estrin, and M. Handley, "Quality Adaptation for Congestion Controlled Video Playback over the Internet," Proc. of ACM SIGCOMM '99, Cambridge, Sept. 1999.
14. Anupama Sundaresan, Gowri Dhandapani, “Diffspec - A Differentiated Services tool”, The University of Kansas Lawrence, KS 66045-2228, December 19, 1999. http://qos.ittc.ukans.edu/DiffSpec/diffspec.html.

Session-Aware Popularity Resource Allocation for Assured Differentiated Services*

Paulo Mendes[1,2], Henning Schulzrinne[1], and Edmundo Monteiro[2]

[1] Department of Computer Science, Columbia University
New York, NY 10027, USA
{mendes,schulzrinne}@cs.columbia.edu

[2] CISUC, Department of Informatics Engineering, University of Coimbra
3030 Coimbra, Portugal,
{pmendes,edmundo}@dei.uc.pt

Abstract. Differentiated Service networks (DS) are fair in the way that different types of traffic can be associated to different network services, and so to different quality levels. However, fairness among flows sharing the same service may not be provided. Our goal is to study fairness between multirate multimedia sessions for an *assured* DS service, in a multicast network environment. To achieve this goal, we present a fairness mechanism called *Session-Aware Popularity Resource Allocation* (SAPRA), which allocates resources to multirate sessions based upon their number of receivers. Simulation results in a multirate and multi-receiver scenario show that SAPRA maximizes the utilization of bandwidth and maximizes the number of receivers with high-quality reception.

Keywords: fairness, multimedia sessions, multicast, differentiated networks, multirate sources.

1 Introduction

Almost all multimedia applications in the Internet use unirate sources, generating flows with rates that don't change over time. For example, the SureStream technology from RealNetworks allows streams' broadcast with multiple rates by creating unirate stream copies. This approach leads to bandwidth waste in heterogeneous environments, such as the Internet, because sources broadcast copies of the same stream in order to satisfy receivers with different quality requirements. This can be solved by replacing unirate sources with multirate ones. Multirate sources [8,19] divide streams into cumulative layers. Each layer has a different rate and importance, and the stream rate is equal to the sum of all its layers' rates. This approach avoids waste of bandwidth, since sources broadcast only one stream to all receivers, sending each stream's layer to a different multicast group. Receivers join as many multicast groups as their connection speed

* This work is supported by POSI-Programa Operacional Sociedade de Informação of Portuguese Fundação para a Ciência e Tecnologia and European Union FEDER

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 74–85, 2002.

allows them [14], starting by the most important layer. We use the designation of *session* to define the group of all layers belonging to the same stream.

Due to their real-time characteristics, multimedia sessions need quality guarantees from the network. These guarantees can be provided by the DS model [2], which allows network providers to aggregate traffic in different services at the boundaries of their network. Each service is based upon a per-hop behavior (PHB), which characterizes the allocation of resources needed to give an observable forwarding behavior (loss, delay, jitter) to the aggregate traffic. One important question about Assured Forwarding (AF) [5] services concerns their capability to be fair. AF services provide *intra-session* fairness, between receivers in the same session, since each session's layer can be mapped to a different drop precedence, considering its importance. However, how to achieve *inter-session fairness* in AF services, allowing receivers from all sessions to get their required quality level without wasting resources, is still a challenging research topic.

The goal of our work is to contribute to the study of *inter-session fairness* between sessions in AF services, keeping the *intra-session fairness* property. To achieve this goal, we propose the enhancement of AF services with a *Session-Aware Popularity Resource Allocation* fair mechanism (SAPRA), which provides *inter-session fairness* by assigning more service bandwidth to sessions with higher number of receivers. SAPRA is a session-based mechanism and not only multicast-based, since hiding session information from DS routers results in *intra-session* unfairness, higher quality oscillations and lower quality for all receivers. SAPRA also includes a resource utilization maximization function, because fairness policies based only upon the number of receivers could still lead to waste of resources. This can occur when the bandwidth assigned to a session is higher than the rate really used by that session, as might happen with mobile phone or personal digital assistant (PDA) sessions, since they have low rate requirements and normally a high number of receivers. SAPRA also detects and punishes high-rate sessions in times of congestion, as an incentive for sessions to adapt to the network capacity.

We present ns[1] simulations that evaluate SAPRA behavior in a multirate multi-receiver environment using a simple dropper, which we called SAPRAD, and using RIO, the dropper normally used in AF.

The remaindder of the paper is organized as follows. In section 2, we present a brief description of some fairness definitions and some multirate source implementations. Section 3 describes SAPRA functionality and section 4 presents simulation results. Finally, section 5 presents some conclusions and future work.

2 Related Work

There are several experimental multirate codecs, such as the Scalable Arithmetic Video Codec from the University of Berkeley[2] developed by D. Taubman [19],

[1] Network Simulator: http://www.isi.edu/nsnam/ns/

[2] Experimental software at: http://www-video.eecs.berkeley.edu

or the Scalable Video Conferencing project from the Framkom Research Corporation[3] [8]. To fairly distribute AF resources between multirate traffic generated by these codecs, the *max-min* fairness definition [1] could be used since its formal definition is a well accepted criterion for fairness and its multicast definition [20] was extended to include multirate sessions [17]. However, Rubenstein et al. [17] show that *max-min* fairness can not be provided in the presence of discrete set of rates, as is the case of multirate sources.

The maximal fairness definition presented by Sankar et al. [18] exists in the presence of a discrete set of rates, but it doesn't consider the number of receivers in each session. Therefore, maximal fairness can't maximize resource utilization and at the same time maximize the number of receivers with good quality level.

Legout et al. present a proposal [11] to distribute bandwidth between sessions considering their number of receivers. However, this proposal assumes that every router in the path between the session's sender and its receivers keep information about the session's layers and the receivers receiving those layers. This proposal also doesn't maximize the utilization of resources and doesn't punish high rate flows.

Li et al. present [12] another proposal to improve *inter-session fairness* based upon the *max-min* fairness definition. Besides *max-min* limitation with discrete multirate sessions, this proposal only considers one shared link and doesn't consider the number of receivers and layers importance of a session.

3 SAPRA Fairness Mechanism

In this section, we introduce the *Session-Aware Popularity Resource Allocation* fairness mechanism (SAPRA), which is implemented only in DS-edge routers.

We assume that each possible multicast branch point is located only in DS-edge routers and that several multimedia applications can share the same host. We name each application *source* and each host *sender*. Since sources are multirate, they generate multimedia sessions with several layers, each layer identified by a Source-Specific Multicast (SSM) channel [6] - sender IP address and destination multicast group. Each receiver can join more than one session at the same time, even if those sessions belong to the same sender. To join a session, receivers start joining the SSM channel of the most important layer. They can try to increase their reception quality by joining more layers, always from the most important one. They can also get information about sessions using, for example, the Session Announcement Protocol (SAP) [4]. The number of receivers in each session correspondes to the number of receivers of the most important layer.

Implementing a fairness mechanism in DS-edge routers that only have information about multicast groups and not about sessions results in intra-session unfairness, higher quality oscillations and lower quality for all receivers. Fig. 1 shows the difference between a scenario where routers have information only

[3] Project page: http://mbc.framkom.se/projects/scale/

about multicast groups and a scenario where routers have knowledge about session.

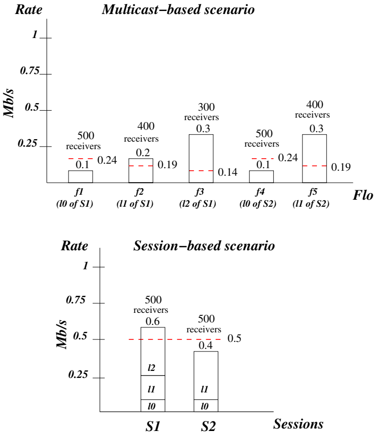


Fig. 1. SAPRA scenarios

We assume that the session-based scenario has two sessions (S_1 and S_2) sharing a link with 1 Mb/s. Each session has 500 receivers, which mean that it has 0.5 Mb/s of bandwidth allocated. Session S_1 has three layers (l_0, l_1 and l_2) joined by 500, 400 and 300 receivers respectively, and session S_2 has two layers (l_0 and l_1) joined by 500 and 400 receivers respectively. In the multicast-based scenario all layers are considered as independent multicast groups (flows f_1 to f_3 are layers from S_1 and flows f_4 and f_5 are layers from S_2), which means that the total number of receivers sharing the link is 2100. Therefore flow f_1 and f_4 have an allocated bandwidth of 0.24 Mb/s each, f_2 and f_5 of 0.19 Mb/s each and f_3 of 0.14 Mb/s. Considering for example session S_1, Fig. 1 shows that the 100 receivers of S_1 that only join l_0 have the same reception rate (0.1Mb/s) and zero loss in both scenarios, since the rate is lower than the fair rate. However the 100 receivers that join l_0 and l_1 have a reception rate of 0.29 Mb/s and 5% loss in the multicast-based scenario and a rate of 0.3 Mb/s and zero loss in the session-based scenario. The situation becomes worst for the 300 receivers that join the three layers, since they have a reception rate of 0.43 Mb/s and 58% losses in the multicast-based scenario and a rate of 0.5 Mb/s and 16% losses in the session-based scenario. This shows that receivers have lower rate and higher loss percentage in a multicast scenario than in a session-based one. The multicast-based scenario isn't also *intra-session fair*, because AF drop precedences don't respect layers' importance. It also presents a higher quality oscillation, since receivers detect losses not only in the less important layer, but also in intermediary ones.

We propose two methods to implement SAPRA as a session-based mechanism in DS-edge routers. In the first method, each sender allocates consecutive multicast addresses to all layers inside a session and keeps one address gap between sessions. With SSM this method doesn't bring any address allocation problem, since each source is responsible for resolving address collisions between all the channels (232/8 addresses) they create. In this scenario each sender manages 2^{24} addresses in IPv4 and 2^{32} per scope in IPv6. With this method, DS-edge routers identify as belonging to the same session all layers that receivers join with consecutive SSM channels. The second proposed method is to change the way IGMPv3 [7] is used. The *auxiliary* data field of IGMPv3 reports can be used to include the multicast address of the most important layer - which identifies the session - in reports about other layers. So, DS-edge routers explicitly know what is the session of each layer. Routers that don't implement SAPRA ignore the *auxiliary* data field as is done in the current IGMPv3 implementations. In both proposed methods, routers know the relationship between layers in a session by the order receivers use to join sessions' layers. Receivers are motivated to join layers from the most important to the less important one, because less important layers are useless without the most important ones in the re-construction of the session's multimedia stream.

We assume that TCP and UDP traffic use different AF services. However unicast and multicast flows can share the same AF service. In this case SAPRA treats unicast flows as sessions with one layer and one receiver only. All layers of the same session use the same AF service, being however marked with different drop precedences. SAPRA only uses two drop precedences, IN and OUT, from the three allowed by AF services. We also assume that sources mark all their traffic as IN.

SAPRA has two components, one agent and one marker. Each DS-edge router has only one SAPRA agent and one marker for each downstream link. SAPRA agents exchange control information periodically with their neighbours. This information includes an *update* message sent to upstream neighbours with the number of receivers and fair rate of each session that presents changes in those values since the last time an *update* message was sent. This reduction of the *update* message size and the fact that agents don't need to have global network knowledge increases SAPRA scalability. The *update* messages information is used by agents to compute sessions' fair rates. Control information also includes a *sync* message sent to downstream neighbours. This message, which contains the lowest fair rate that each session has in the path from the source, can be used by quality adaptive mechanisms in the receivers. A brief description of the protocol used to exchange *update* and *sync* messages is presented in [16] and its performance study will be presented in a future paper. Next, we describe the SAPRA agent and marker.

3.1 SAPRA Agent

When a SAPRA agent receives an *update* message, it updates the local information about the sessions in the message and computes their new fair rates. In DS-edge routers that have local receivers, agents gather the number of receivers from IGMPv3 "State-Changes" reports.

Agents have to reserve local resources to store the received and computed information. For each upstream interface, agents reserve four bytes for each session and four bytes for each layer. For the local interface and for each downstream interface agents reserve twelve bytes for each session and eight bytes for each layer. As an example, consider 1000 sessions, each one with three layers, that are going through a DS-edge router with three downstream interfaces. Consider also that each session is present in each downstream interface and that the router doesn't have local receivers. In this situation the router reserves 124 Kb.

To compute session S_u fair rate, F_{ui}, in a link i, agents use Eq. 1, which defines F_{ui} as the ratio between the session's number of receivers, n_{ui}, and the total number of receivers in that link, considering the AF service capacity[4], C_i. In Eq. 1 m_i is the number of sessions that share link i.

$$F_{ui} = \left(\frac{n_{ui}}{\sum_{x=1}^{m_i} n_{xi}}\right) * C_i \tag{1}$$

All computed fair rates are adjusted considering downstream fair rates. This adjustment is required to maximize the utilization of resources, because sessions whose fair rate is higher than their downstream fair rate waste resources, since packets are dropped downstream. Therefore, if a session has a computed fair rate, F_{ui}, higher than its downstream fair rate, F_{uj} (j is a link downstream of i), F_{ui} becomes equal to F_{uj} and the rate difference, $F_{ui} - F_{uj}$, is added to the available shared bandwidth in the link, w_i. The available shared bandwidth allows fair rate increase for sessions that have a fair rate lower than their downstream fair rate. If the difference $F_{uj} - F_{ui}$ is lower than w_i, then F_{ui} becomes equal to F_{uj} and w_i is reduced by that difference. However if $F_{uj} - F_{ui}$ is higher than w_i, then F_{ui} is only added by w_i, and w_i becomes zero.

The available shared bandwidth is used by all sessions, starting by those with the highest number of receivers. This maximizes the utilization of resources increasing the number of receivers with good quality.

Agents functionality can be described by a fairness definition, which can be stated as: Consider that F_{ui} and F_{uj} are the fair rates of a session S_u in a link i and in a link j downstream of i, respectively. A fair rate allocation vector $V_i^1(F_{1i}^1, \ldots, F_{m_i}^1)$ in a link i is said to be *SAPRA*-fairer if for any alternative feasible[5] fair rate allocation vector $V_i^2(F_{1i}^2, \ldots, F_{m_i}^2)$:

$$\forall u \in [1, m_i],\, F_{ui}^2 > F_{ui}^1 \wedge F_{ui}^2 \le F_{uj}^2 \;\Rightarrow\; \exists v \in [1, m_i],\, F_{vi}^2 < F_{vi}^1 \wedge F_{vi}^1 \le F_{vj}^1 \tag{2}$$

After being adjusted, sessions' fair rates are passed by the agent to each SAPRA marker present in the downstream links.

3.2 SAPRA Marker

Fig. 2 shows the SAPRA marker - shadowed component - which replaces the usual marker in AF services.

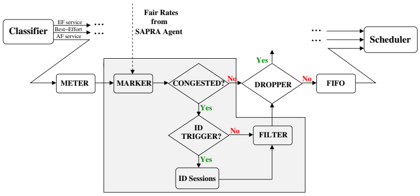


Fig. 2. SAPRA marker

This enhances the AF service with the capability to fairly distribute resources between sessions, based upon the fair rates computed by the SAPRA agent and the layers' average rate.

The marker needs to know the arrival rate of each layer. The easiest way to achieve this would be to obtain that information directly from the sources. However sources could indicate a lower rate than they actual have, trying to get a higher percentage of IN packets. Therefore a meter, included in the DS model

[4] How the AF capacity in a DS-edge router is configured is a DS model implementation concern.

[5] A feasible vector means that the sum of all fair rates is equal or lower than the AF capacity.

as shown in Fig. 2, is used to estimate average rates, maintaining the fairness mechanism independent of the sources.

With the information from the agent and the meter, the marker marks layer traffic as IN or OUT. Only packets that arrive already marked IN will be remarked, since OUT packets are not compliant with upstream fair rates. All incoming IN packets are marked IN or OUT as follows: Considering that l_0 is the most importance layer and l_n the less important one of a session S_u, Eq. 3 and Eq. 4 give the probability that a layer l_k of that session has to be marked IN, P_{uki}^{in}, and OUT, P_{uki}^{out}, in a link i. With this marking strategy there is also a differentiation between sessions that have traffic marked OUT, since sessions with higher rates will have more packets marked OUT.

$$P_{uki}^{in} = \begin{cases} 1 & , L_{uki} \leq F_{ui} \\ \frac{M_{uki} - L_{u(k-1)i}}{r_{uki}^{in}} & , L_{uki} > F_{ui} \end{cases} \tag{3}$$

$$P_{uki}^{out} = \begin{cases} 0 & , L_{uki} \leq F_{ui} \\ \frac{L_{uki} - M_{uki}}{r_{uki}^{in}} & , L_{uki} > F_{ui} \end{cases} \tag{4}$$

In Eq. 3 and Eq. 4 session S_u has the following values in link i: fair rate F_{ui}; rate of IN packets of its layer l_k, r_{uki}^{in}; sum of all rates from layer l_0 to layer l_k- $L_{uki} = \sum_{x=0}^{k} r_{u,x,i}^{in}$ -; maximum value between the session's fair rate and the sum of all layers' rate from l_0 to l_{k-1} - $M_{uki} = max(F_{ui}, L_{u(k-1)i})$.

When the meter detects that the link is congested, the marker filters all layers from sessions with rate higher than their fair rate plus their share of the available bandwidth, before sending the marked packets to the DS dropper. We define these sessions as *high-rate sessions*. The strategy to identify and punish high-rate sessions is based upon the Random Early Detection with Preferential Dropping mechanism (RED-PD) [13]. However, contrary to RED-PD, SAPRA uses fixed length intervals in congested periods to identify high-rate sessions and doesn't need to maintain a list of all layers that suffer drops in each interval. This simplifies the mechanism avoiding the estimation of the recent average packet drop rate used by RED-PD to compute their variable interval length.

In each identification interval, SAPRA starts by verifying which sessions have total (IN and OUT packets) rate, r_{ui}, higher than their fair rate. Session S_u total rate in a link i, is given by $r_{ui} = \sum_{x=0}^{n-1} r_{uxi}$, considering that each layer l_k from l_0 to l_{n-1} has rate r_{uki}.

A session with rate lower than its fair rate isn't using all its share of the link bandwidth, so the unused bandwidth becomes available for other sessions' OUT packets. Fig. 3 shows that SAPRA distributes this available bandwidth in equal shares between all sessions with rate higher than their fair rate and identifies which of these sessions are high-rate sessions.

To punish each high-rate session S_u in a link i, SAPRA computes its dropping probability in each identification interval t, $D_{ui}(t)$, using Eq. 5, where z_i is the available bandwidth in link i.

$$D_{ui}(t) = (D_{ui}(t-1) + \sigma_d + \frac{1}{100} * (1 - \frac{F_{ui} + \frac{z_i}{m_i}}{r_{ui}})) \tag{5}$$

This equation shows that in each interval the dropping probability of high-rate sessions is increased by two values: a drop factor, σ_d, and a value proportional to the excess rate the session is using. This excess rate corresponds to the difference between the session rate and the sum of the session fair rate and its share of the available bandwidth, as shown in Fig. 3.

The dropping probability of each session is used to compute dropping probabilities for their layers, being the less important layer the first to suffer an increase of its dropping, because losses induce a higher quality degradation if they happen in more important layers [9]. Since hierarchical codecs are tolerant to loss in less important layer, SAPRA computes layers' dropping probability with a linear quality degradation. However SAPRA can be configured to be more aggressive, dropping all layers that have a dropping probability higher than a predefined limit Θ_d.

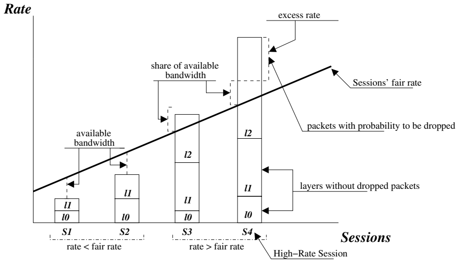


Fig. 3. Punishment mechanism

If in an identification interval a session is no longer identified, its dropping probability is halved until it reaches a minimum value, after which the session will stop to be filtered.

Packets that aren't dropped by the filter are sent to the DS dropper as happens with all packets from non-identified sessions. In the DS model the dropper is managed with RIO (RED with in/out) [3]. However, RIO introduces some complexity, since it needs to compute the total average queue size, the average queue size of IN packets, has different dropping scheme (random, front and tail) and its four thresholds can introduce oscillations. Therefore, we show that SAPRA has similar behavior with RIO and with a simpler dropper, which we named SAPRAD (SAPRA Dropper). SAPRAD manages a FIFO queue preferentially dropping OUT packets. When the queue is full an OUT packet is randomly discarded. Only if the queue doesn't have OUT packets, an IN packet is randomly discarded. This guarantees that layers with higher rates are more severely punished.

4 SAPRA Simulations

In this section we present simulations that aim to analyse the ability of SAPRA to distribute AF bandwidth between multirate sessions with different number of receivers, considering the number of receivers and the relationship between layers inside each session. We use a scenario - Fig. 4 - with three DS-edge routers and two congested links. The upstream link is configured with 10 Mb/s and the downstream with 5 Mb/s of bandwidth. The queue in each link has a size of 64 packets - default value in Cisco IOS 12.2 -, and each packet has a size of 1000 bytes. We analyse SAPRA's behavior in the presence of two types of droppers, RIO, the dropper normally used in AF, and SAPRAD.

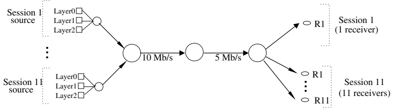


Fig. 4. Simulation scenario

In these simulations we use 11 sessions, S_1 to S_{11}, each one with three layers, l_0, l_1 and l_2, being l_0 the most important. Each layer is identified by a SSM channel and each session has a different number of receivers, from one in $S_1$to eleven receivers in S_{11}, increasing by one receiver per session. Although SAPRA can deal with any number of layers, we consider sessions with three layers in the present simulations, since this partitioning provides a good quality/bandwidth trade-off and additional layers only provide marginal improvements [9]. We performed sixty seconds simulations with sources that have increasing rates, from $S_1$to S_{11}, in multiples of 25 Kb/s from session to session, starting with 25 Kb/s for S_1. The session rate is the rate of its most important layer, l_0, and each layer l_k has a rate equal to twice the rate of l_{k-1}. The dropping probability of all sessions is computed using Eq. 5, with σ_d equal to 0.5% and the dropper used is SAPRAD.

Fig. 5 shows that sessions' fair rates are proportional to sessions' number of receivers, since SAPRA distributes resources considering the number of receivers in each session. They also show that, in the upstream link, sessions use fair rates lower than the computed ones. This happens because SAPRA adjusts the upstream link computed fair rates, since they are higher than the downstream link ones. Another conclusion is that SAPRA respects layers' relationship, since packet dropping starts always by l_2, which can be clearly seen in Fig. 5 (right) where the rate of l_2 is reduced to a minimum value.

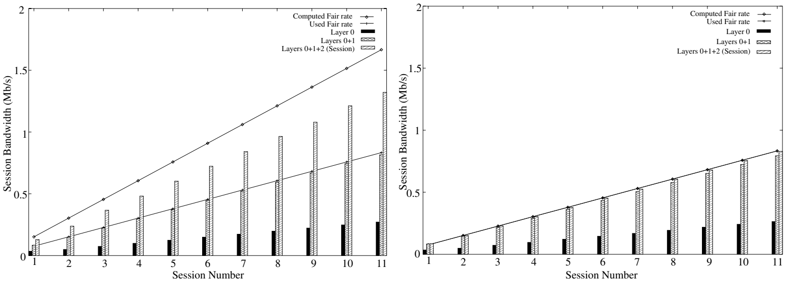


Fig. 5. Sessions' fair rates in upstream (left) and downstream (right) links

In Fig. 5 sessions with higher rate suffer higher drop rates. For example, in the upstream link, S_{11} incoming rate is 1925 Kb/s and it has a loss rate of 31.35%, while S_{10} has an incoming rate of 1213 Kb/s and 30.72% loss. Fig. 5 also shows that in the upstream link all sessions are identified as high-rate sessions, since their incoming rates are higher than their fair rates and therefore the available bandwidth is zero.

The filter agressiveness can be configured by changing σ_d and the dropping probability limit Θ_d. To better show the punishment effect, we used an equal number of receivers per session, i.e., sessions have the same fair rate. Fig. 6 shows results for the upstream link using a σ_d value of 5%: In the left figure, Θ_d isn't defined and so layers suffer a linear dropping increase and in the latter; in the right one, Θ_d is equal to 50%, after which layers are completely dropped.

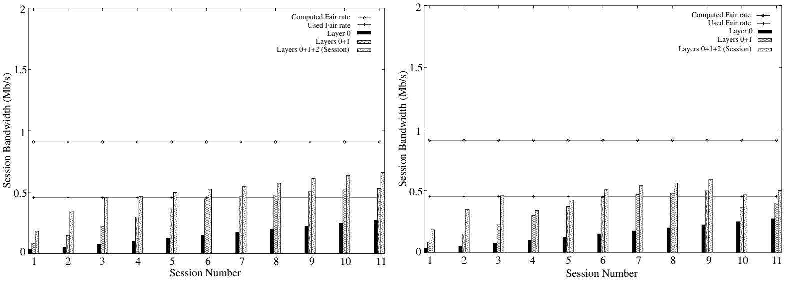


Fig. 6. Punishment mechanism with σ_d value of 5%

Fig. 6 shows that all layers in S_1 and S_2 have null dropping probability. The same happens for l_0 in any session. As for l_2 - Fig. 6 (right) - in S_3 presents a dropping probability of 22%, in S_4 of 61%, growing up until 100% from S_7 to S_{11}, which means that it's completely dropped from S_4 to S_{11}. Nevertheless, l_2 doesn't have a null rate in these sessions. This is due to the time gap between the beginning of the simulation and the moment agents receive the first *update* message, during which agents don't have any information about sessions, being unable to differentiate them. Consequently all layers have the same dropping probability, making possible for receivers to get a certain number of packets from all layers. SAPRA has similar results for the downstream link. These results can be found in [15].

To compare SAPRAD behavior against RIO, we used again a sixty seconds simulation but sessions with equal rate and one receiver only. SAPRA uses a value of 0,5% for σ_d and Θ_d isn't defined. RIO's minimum and maximum thresholds have the following values: IN_min of 60 packets, IN_max of 64 packets, IN_drop of 0.5%, OUT_min of 32 packets, OUT_max of 48 packets and OUT_drop of 50%. With these values the dropping of OUT packets is higher than the one of IN packets for the 64 packets queue, which approximates the behavior of the SAPRAD dropper.

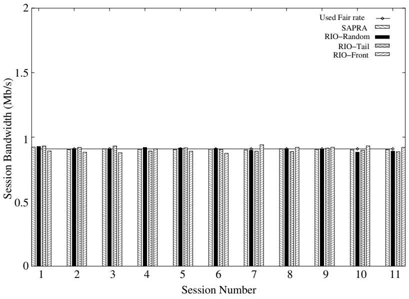


Fig. 7. SAPRAD and RIO

Fig. 7 shows, for the upstream link, that SAPRAD and RIO behavior is similar, being the session rate closer to its fair rate in the link when SAPRAD is used. These results show that an Internet Service Provider (ISP) that uses RIO can still use SAPRA to distribute AF resources between multimedia sessions. However implementing SAPRA with SAPRAD instead of RIO reduces the mechanism complexity. To understand what is the best RIO's configuration we made several simulations by changing the value of IN and OUT thresholds. These simulations show that SAPRA behavior with RIO has a high variation between different RIO configurations. Detailed results can be found in [15].

5 Conclusion and Future Work

This paper describes and evaluates SAPRA, whose components are only installed in DS-edge routers, computing sessions' fair rate based upon SAPRA fairness definition. SAPRA enhances DS functionality by fairly distributing bandwidth and by punishing high-rate sessions.

SAPRA distributes bandwidth between sessions considering their number of receivers, which increases receivers motivation to use multicast, since they will experience higher quality than unicast ones. SAPRA also increases providers' motivation to use multicast: ISPs can have more clients using fewer resources and multimedia providers can deploy new services that scale with large number of receivers. However, SAPRA doesn't attempt to be the optimal fairness mechanism, because social and economic issues can influence fairness as much as technical ones. But being based upon sessions' number of receivers and a maximal resource utilization function, SAPRA can be the base of a hierarchical fairness mechanism for multirate multicast sessions.

To evaluate SAPRA behavior, we presented simulations with two congested links that showed its performance with a simple dropper, SAPRAD, and also with RIO. Simulations showed that SAPRA maximizes the utilization of bandwidth and the number of receivers with high quality reception.

As future work we'll simulate SAPRA in more complex scenarios in order to analyze the SAPRA protocol oscillations with the variation of the number of receivers. We'll also create a receiver-driven adaptive mechanism that will use SAPRA network support, mainly fair rates collected in *sync* messages, trying to solve some of the problems presented by other adaptive mechanisms such as RLM [14] and RLC [21]. Legout et al. [10] show that RLM presents *inter-session* unfairness and has low convergence time and low link utilization, while RLC is unfair to TCP for large packets and its bandwidth inference mechanism is very sensitive to queue size. Also, both mechanisms can induce losses in all layers when a join experience occurs. This can be avoided if the adaptive mechanim is based upon SAPRA, since it guarantees *intra-session* fairness.

References

1. D. Bertsekas and R. Gallager. *"Data Networks"*. Prentice-Hall, 1987.

2. S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. “An architecture for differentiated service”. Request for Comments 2475, Internet Engineering Task Force, December 1998.
3. D. Clark and W. Fang. “Explicit allocation of best-effort packet delivery service”. *Journal of IEEE/ACM Transactions on Networking*, 6(4):362–373, August 1998.
4. M. Handley, C. Perkins, and E. Whelan. “Session announcement protocol”. Request for Comments 2974, Internet Engineering Task Force, October 2000.
5. J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski. “Assured forwarding PHB group”. Request for Comments 2597, Internet Engineering Task Force, June 1999.
6. H. Holbrook and B. Cain. “Source-specific multicast for IP”. Internet draft, Internet Engineering Task Force, March 2001.
7. H. Holbrook and B. Cain. “Using IGMPv3 for source-specific multicast”. Internet draft, Internet Engineering Task Force, March 2001.
8. M. Johanson. “Scalable video conferencing using subband transform coding”. In *In Proc. of ICSPAT’99*, Orlando, FL, USA, November 1999.
9. J. Kimura, F. Tobagi, J. Pulido, and P. Emstad. “Perceived quality and bandwidth characterization of layered MPEG-2 video encoding”. In *In Proc. of SPIE International Symposium on Voice, Video and Data Communications*, Boston, MA, USA, September 1999.
10. A. Legout and E. W. Biersack. “Pathological behaviors for RLM and RLC”. In *In Proc. of NOSSDAV’00*, Chapel Hill, NC, USA, June 2000.
11. A. Legout, J. Nonnenmacher, and E. W. Biersack. “Bandwidth allocation policies for unicast and multicast flows”. In *In Proc. of IEEE INFOCOM’99*, New York, NY, USA, March 1999.
12. X. Li, S. Paul, and M. Ammar. “Multi-session rate control for layered video multicast”. In *In Proc. of Multimedia Computing and Networking*, San Jose, CA, USA, January 1999.
13. R. Mahajan and S. Floyd. “Controlling high-bandwidth flows at the congested router”. Tr-01-001, ICSI, April 2001.
14. S. McCanne, V. Jacobson, and M. Vetterli. “Receiver-driven layered multicast”. In *In Proc. of ACM SIGCOMM’96*, pages 117–130, Palo Alto, CA, USA, August 1996.
15. P. Mendes. “Session-aware popurality resource allocation”. `http://www.cs.columbia.edu/~mendes/sapra.html`.
16. P. Mendes, H. Schulzrinne, and E. Monteiro. “Multi-layer utilization maximal fairness for multi-rate multimedia sessions”. Cucs-007-01, Columbia University, July 2001.
17. D. Rubenstein, J. Kurose, and D. Towsley. “The impact of multicast layering on network fairness”. In *Proc. of ACM SIGCOMM’99*, Cambridge, MA, USA, September 1999.
18. S. Sankar and L. Tassiulas. “Fair allocation of discrete bandwidth layers in multicast networks”. In *Proc. of IEEE INFOCOM’00*, Tel Aviv, Israel, March 2000.
19. D. Taubman and A. Zakhor. “Multirate 3-D subband coding of video”. *Journal of IEEE Transactions on Image Processing*, 3(5):572–588, September 1994.
20. H. Tzeng and K. Siu. “On max-min fair congestion control for multicast ABR service in ATM”. IEEE Journal on Selected Areas in Communications, 15:542–556, April 1997.
21. L. Vicisano, L. Rizzo, and J. Crowcroft. “TCP-Like Congestion control for layered multicast data transfer”. In *Proc. of IEEE INFOCOM’98*, San Francisco, CA, USA, March/April 1998.

Most Probable Path Techniques for Gaussian Queueing Systems

Ilkka Norros

VTT Information Technology, P.O. Box 1202, 02044 VTT, Finland

Abstract. This paper is a review of an approach to queueing systems where the cumulative input is modelled by a general Gaussian process with stationary increments. The examples include priority and Generalized Processor Sharing systems, and a system where service capacity is allocated according to predicted future demand. The basic technical idea is to identify the most probable path in the threshold exceedance event, or a heuristic approximation of it, and then use probability estimates based on this path. The method is particularly useful for long-range dependent traffic and complicated traffic mixes, which are difficult to handle with traditional queueing theory.

1 Introduction

This paper is a review of an approach to queueing systems with Gaussian input. The motivation to study such systems is twofold. On one hand, complicated dependence structures are easiest to study first in a Gaussian framework, where the dependence is reduced to correlation. This is also the historical origin of this work — it started with queues with fractional Brownian motion (fBm) as input [19], which is the simplest process that has the self-similarity property, first observed in the famous Bellcore measurements [11]. On the other hand, it could be expected that, thanks to the Central Limit Theorem, traffic in high capacity systems would be rather well modelled with Gaussian processes [1]. Empirical studies indicate, however, that a good fit to Gaussian distribution may require very high traffic aggregation levels. The Gaussian approach can be useful in making rough performance estimates for Differentiated Services in Internet, because one works there with large traffic aggregates.

Our interest in most probable paths started by applying the generalized Schilder's theorem to the fBm queue [20]. The approach was extended to ordinary queues with general Gaussian input in [2,3], and further to priority queues in [14]. In [13] and [15], we applied a similar machinery to Generalized Processor Sharing (GPS) schedulers and presented a somewhat improved version of the priority case. Most of this research was done within the COST Actions 257 and 279. A summary on Gaussian traffic modelling, linked to the technical documents, can be found in the hypertext Final Report of the action [26].

The paper is structured as follows. We start with discussing the definitions of Gaussian queueing systems in Section 2. This involves some technical details caused by the unavoidable presence of negative traffic in Gaussian modelling.

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 86–104, 2002.

Section 3 presents the main ideas of our approach. A central role is played by the most probable paths along which queue size thresholds are exceeded. The rest is devoted to two cases, where the most probable paths obtain particularly interesting shapes. Section 4 shows how the most probable path can experience a kind of "phase transition" between short and long busy periods. Section 5 studies a simple model of dynamical capacity allocation. This is a new type of application, first time presented here.

2 Definition of Gaussian Queueing Systems

2.1 Gaussian Models of Traffic

Our basic traffic model is a continuous Gaussian process $A = (A_t)_{t\in\Re}$ with stationary increments. For $s < t$, $A_t - A_s$ presents the amount of traffic in time interval $(s, t]$, and we set $A_0 \equiv 0$. A process is called Gaussian, if all its finite-dimensional distributions are multivariate Gaussian. The property of stationary increments means that for any $t_0 \in \Re$, the processes A and $(A_{t+t_0} - A_{t_0})_{t\in\Re}$ have the same finite-dimensional distributions.

We denote $A(s, t) = A_t - A_s$, and use similar notation for other processes as well.

The use of Gaussian models for big traffic aggregates can be justified by the Central Limit Theorem. However, even the question about the Gaussian character of some traffic cannot be raised without specifying the relevant timescale, say δ. There should be a large number of individual sources contributing to the traffic in every time interval of size δ. Moreover, if the marginal distribution of the contribution of an individual source in those intervals has very high variability, the application of CLT may still be problematic. In our study on Internet users over ISDN [10], it was found that a few Mbit/s of such traffic had good fit with Gaussian distribution when the time resolution δ was coarser than 100 ms. Note that this traffic was exceptionally well-behaving, because the users were restricted to the ISDN access speed.

A non-pleasant special feature of Gaussian models is that there is always a positive probability of negative input. Such input does not correspond to anything real, and its existence destroys some classical arguments of queueing theory. In a Gaussian framework, the non-problematic definitions of queueing theory must be replaced by analogously defined functionals of a Gaussian process. Moreover, we don't have much hope to obtain other kinds of rigorous general results on the distributions of these functionals than inequalities and limit theorems. At our present "state-of-art", we must often be satisfied with heuristic approximations.

Despite these reservations, Gaussian models are tempting because of their many nice features:

- a Gaussian process with stationary increments is completely characterized by its mean $m = \mathbb{E}\{A_1\}$ and cumulative variance function $v(t) = \mathrm{Var}(A_t)$; indeed, we can write
$$A_t = mt + Z_t,$$

where Z is a centered (mean zero) process, and the covariance function of A (and Z) can be written as

$$\mathrm{Cov}\,(A_s, A_t) = \mathrm{Cov}\,(Z_s, Z_t) = \frac{1}{2}(v(s) + v(t) - v(s-t));$$

- a superposition of independent Gaussian traffic streams is Gaussian;
- multiclass traffic consisting of Gaussian traffic classes, such that their joint distribution is Gaussian also, can be studied within the same framework;
- unlike most other traffic models, A_t has an explicitly known (Gaussian) distribution for any t;
- the quantities m and $v(t)$ can be rather well estimated from measurement data;
- long-range dependence does not provide any extra difficulty.

In the multiclass case, let the input traffic consist of k classes, and denote the cumulative arrival process of class $j \in \{1, \ldots, k\}$ by $(A_t^{\{j\}})_{t\in\Re}$. We also denote $A^{\{j\}}(s,t) \doteq A_t^{\{j\}} - A_s^{\{j\}}$. For the superposition of a set of traffic classes $J \subseteq \{1, \ldots, k\}$ we write

$$A_t^J \doteq \sum_{j\in J} A_t^{\{j\}}$$

and use similar superscript notation also for other quantities defined later. We assume that the processes $A^{\{j\}}$ are independent, continuous Gaussian processes with stationary increments and denote

$$A_t^{\{j\}} = m_j t + Z_t^{\{j\}}, \quad m = \sum_{i=1}^{k} m_i, \quad \mathrm{Var}\left(Z_t^{\{j\}}\right) = v_j(t), \tag{1}$$
$$\Gamma_j(s,t) = \mathrm{Cov}\left(Z_s^{\{j\}}, Z_t^{\{j\}}\right),$$

where the $Z^{\{j\}}$'s are centered (zero-mean) processes. To exclude certain degenerate cases, we assume that

$$\exists \alpha \in (0,2) : \lim_{t\to\infty} \frac{v_i(t)}{t^\alpha} = 0, \quad i \in \{1, \ldots, k\}. \tag{2}$$

Finally, let us specify the mathematical framework completely. Define a path space Ω_1 as

$$\Omega_1 = \left\{\omega : \ \omega \text{ is continuous } \Re \to \Re, \ \omega(0) = 0, \ \lim_{t\to\pm\infty} \frac{\omega(t)}{1+|t|} = 0\right\}.$$

(The relation $\lim_{t\to\infty} Z_t^{\{i\}}/t = 0$ a.s., is a consequence of (2) — see [3].) Equipped with the norm

$$\|\omega\|_{\Omega_1} = \sup\left\{\frac{\omega(t)}{1+|t|} \ : \ t \in \Re\right\},$$

Ω_1 is a separable Banach space. We choose $\Omega = \Omega_1^k$ as our basic probability space by letting P be the unique probability measure on the Borel sets of Ω such that the random variables $Z_t^{\{i\}}(\omega_1, \ldots, \omega_k) = \omega_i(t)$ form independent Gaussian processes with covariance functions $\Gamma_i(\cdot,\cdot)$.

2.2 Definition of Simple Queues

Consider first the case of a simple queue, i.e. $k = 1$, and let the server have a constant capacity c. The storage process (queue length process) is then naturally defined as

$$Q_t = \sup_{s \le t}(A(s,t) - c(t-s)). \tag{3}$$

The process Q is obviously stationary, and a sufficient stability condition is that $m < c$.

Because only the net input process $A_t - ct$ matters in this definition, it can be extended to the case that the service process is stochastic as well. Indeed, assume that the cumulative service capacity process C_t is a Gaussian process with stationary increments such that the difference $A_t - C_t$ is also Gaussian with stationary increments and a negative mean rate. The queue length process is then

$$Q_t = \sup_{s \le t}(A(s,t) - C(s,t)), \tag{4}$$

and all results for simple Gaussian queues are applicable. One example of this is given in Section 5.

2.3 Definitions of GPS and Priority Queues

The Generalized Processor Sharing (GPS) service discipline [23] (an idealized version of Weighted Fair Queueing) is a theoretical model which isolates flows and provides service differentiation. Let us consider a GPS queueing system for our k traffic classes, such that the guaranteed service rate for each class i is $\mu_i c$, where $c > m = \sum_i m_i$, $\mu_i > 0$ for each i, and $\sum \mu_i = 1$.

It is not at all obvious how a GPS queue should be defined when negative input is allowed. An elegant definition which results in positive queue length processes even in our case was given by Massoulie [16]. Assume that the amount of potential service for each class i in time interval (s,t) is $\mu_i cT(s,t)$, where $T(s,t) = T_t - T_s$ and T is a non-decreasing stochastic process with $T_0 \equiv 0$. T varies according to the number of backlogged classes. The queue of class i, $Q^{\{i\}}$, and the total queue Q then satisfy

$$\begin{aligned} Q_t^{\{i\}} &= \sup_{s \le t}(A^{\{i\}}(s,t) - \mu_i cT(s,t)) \\ Q_t &= \sup_{s \le t}\left(\sum_{i=1}^{k} A^{\{i\}}(s,t) - c(t-s)\right). \end{aligned} \tag{5}$$

Together with the requirement $Q_t = \sum_{i=1}^{k} Q_t^{\{i\}}$, the equations (5) uniquely define the $k+1$ processes $Q^{\{1\}}, \ldots, Q^{\{k\}}$ and Q_t [16]. The construction works and yields non-negative queues in the Gaussian case also.

Let us then turn to priority queues. Assume that there are k priority classes, numbered with descending priority. There is no distinction between preemptive and non-preemptive priority, because the model is continuous. Since lower class traffic does not disturb upper class traffic, a simple approach is the following:

define $Q^{\{1\}}$, $Q^{\{1,2\}}$, $Q^{\{1,2,3\}}$ etc. as ordinary queues with service rate c, and then set

$$\begin{aligned} Q^{\{2\}} &= Q^{\{1,2\}} - Q^{\{1\}}, \\ &\cdots \\ Q^{\{k\}} &= Q^{\{1,\ldots,k\}} - Q^{\{1,\ldots,k-1\}}. \end{aligned} \tag{6}$$

Using this definition with Gaussian traffic has the non-desirable effect that it does not yield non-negative queue lengths to other classes than the first one. This has, however, little significance in the cases where Gaussian modeling is adequate, so we prefer using it. (Massoulie's GPS definition does not work, as such at least, with $\mu_2 = 0$, which would correspond to a two-class priority queue. It is shown in [15] how discrete time Gaussian priority queues can be defined in such a way that the individual queues are non-negative and sum up to the total queue, and the continuous time could probably be obtained as a limit when the discretization step goes to zero.)

3 Probability Estimates Based on Most Probable Paths

3.1 The Reproducing Kernel Hilbert Space and Large Deviations of Gaussian Processes

For $i = 1, \ldots, k$, the reproducing kernel Hilbert space (RKHS) R_i of the process $Z^{\{i\}}$ is defined as follows (see, e.g., [4]): start with the functions $\Gamma_i(t, \cdot)$, $t \in \Re$, define their inner products as

$$\langle \Gamma_i(s, \cdot), \Gamma_i(t, \cdot) \rangle_{R_i} \doteq \Gamma_i(s, t),$$

extend to a linear space (with pointwise operations), and complete the space with respect to the norm $\|f\|_{R_i} \doteq \langle f, f \rangle_{R_i}$. It is easy to verify that R_i is a linear subspace of Ω_1, and the topology induced by $\|\cdot\|_{R_i}$ is finer than that induced by $\|\cdot\|_{\Omega_1}$.

The RKHS of the multivariate process $(Z_t^{\{1\}}, \ldots, Z_t^{\{k\}})$ is, by the independence of the $Z^{\{i\}}$'s, $R \doteq R_1 \times \cdots \times R_k$ with the inner product

$$\langle (f_1, \ldots, f_k), (g_1, \ldots, g_k) \rangle_R \doteq \sum_{i=1}^{k} \langle f_i, g_i \rangle_{R_i}.$$

The *reproducing kernel property*, which is a straightforward consequence of the definition of the inner products, tells that within R, the functions can be evaluated by taking an inner product with a corresponding vector of covariance functions:

$$\langle (f_1, \ldots, f_k), (\Gamma_1(t_1, \cdot), \ldots, \Gamma_k(t_k, \cdot)) \rangle_R = \sum_{i=1}^{k} f_i(t_i). \tag{7}$$

The above construction can be further extended to the case that the component processes are dependent, as long as all joint distributions are Gaussian.

A large deviation principle for Gaussian measures in Banach space is given by the generalized Schilder's theorem (Bahadur and Zabell [6], see also [5,9]).

Theorem 1. *The function* $I : \Omega \to \Re \cup \{\infty\}$,

$$I(\omega) = \begin{cases} \frac{1}{2}\|\omega\|_R^2, & \text{if } \omega \in R, \\ \infty, & \text{otherwise,} \end{cases}$$

is a good rate function for the centered Gaussian measure P, and P satisfies the following large deviation principle:

$$\text{for } F \text{ closed in } \Omega: \quad \limsup_{n\to\infty} \frac{1}{n} \log \mathbb{P}\left(\frac{Z}{\sqrt{n}} \in F\right) \le - \inf_{\omega \in F} I(\omega);$$

$$\text{for } G \text{ open in } \Omega: \quad \liminf_{n\to\infty} \frac{1}{n} \log \mathbb{P}\left(\frac{Z}{\sqrt{n}} \in G\right) \ge - \inf_{\omega \in G} I(\omega).$$

Thus, the essential problem is to find a path ω that minimizes $I(\omega)$ in a given set B, or, equivalently, the norm $\|f\|_R$ in the set $B \cap R$. We call it the *most probable path* in that set. Intuitively, one can can think of $e^{-I(\omega)}$ as something like the probability density of our infinite dimensional Gaussian measure, so that minimizing $I(\omega)$ corresponds to maximizing likelihood. In most cases, the most probable path is unique, but the examples in Section 4 show that non-unique paths may appear and even have interesting meaning as "phase transitions" of the queueing system.

The approach presented here was originally motivated by the generalized Schilder's theorem [20]. However, our main interest is not in large deviations limits but in estimates that are applicable for whole distributions. It was shown in [2] by examples of ordinary queues that estimates of the type $\mathbb{P}(A) \approx \exp(-\inf_{\omega\in A} I(\omega))$ give indeed often a reasonable approximation of the whole queue length distribution, not only for tail behavior. On the other hand, note that it is problematic to even formulate large deviations limit theorems with Gaussian traffic, because the Gaussian character is already the result of another kind of limit procedure, the Central Limit Theorem.

3.2 Half-Space Approximations

Consider first the case of a simple queue. What can be said about the marginal distribution of Q_t? Writing (cf. [17])

$$\{Q_t > x\} = \left\{\sup_{s\le t} \frac{Z_t - Z_s}{x + (c-m)(t-s)} > 1\right\}$$

we see that this event is in fact of the form $\left\{\sup_s Y_s^{(x,t)} > 1\right\}$ for the *centered* Gaussian process $Y_s^{(x,t)} = (Z_t - Z_s)/(x + (c-m)(t-s))$. Thus, we encounter the very classical problem of estimating the distribution of the maximum of a centered Gaussian process. Consider the obvious lower bound

$$\mathbb{P}(Q_t > x) \ge \sup_{s\le t} \mathbb{P}\left(Y_s^{(x,t)} > 1\right) = \overline{\Phi}\left(\frac{x + (c-m)u^*}{\sqrt{v(u^*)}}\right) = \ell(x), \qquad (8)$$

where $\overline{\Phi}$ is the residual distribution function of the standard normal distribution and $u^* > 0$ minimizes $(x + (c - m)u)^2/v(u)$ w.r.t. u. The value u^* has the important practical meaning of characterizing the *relevant timescale* of queues of length x.

Note the geometry of the set $\{Q_t > x\}$: it is the union over s of the sets $\{A(t - s) - c(t - s) > x\}$ which are half-spaces, and thus the complement of a convex set containing the origin. Let f^* be a most probable path in $\{Q_t > x\}$. The following proposition, which we formulate directly in the multiclass case, gives an explicit expression of f^*.

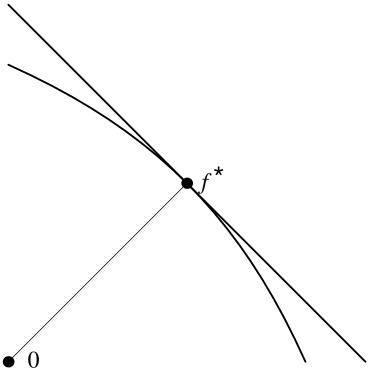


Fig. 1. The half-space $\{-Z_{-t^*} \geq x + (c - m)t^*\}$ is contained in the set $\{Q_0 \geq x\}$. For both sets, the closest point to origin is f^*.

Proposition 1. *Most probable path vectors f^* in the set $\left\{Q_0^{\{1,\ldots,k\}} \geq x\right\}$ have the form*

$$-\frac{x + (c - m)t^*}{\sum_{i=1}^k v_i(t^*)}(\Gamma_1(-t^*, \cdot), \ldots, \Gamma_k(-t^*, \cdot)),$$

where $t^* > 0$ minimizes the expression

$$h(t) = \frac{(x + (c - m)t)^2}{\sum_{i=1}^k v_i(t)}. \tag{9}$$

Proof. Note that

$$\begin{aligned}\left\{Q_0^{\{1,\ldots,k\}} \geq x\right\} &= \bigcup_{s\leq 0} \left\{A^{\{1,\ldots,k\}}(s, 0) - c(0 - s) \geq x\right\} \\ &= \bigcup_{s\leq 0} \left\{Z^{\{1,\ldots,k\}}(s, 0) \geq x + (c - m)(0 - s)\right\},\end{aligned}$$

and, by the reproducing kernel property,

$$\begin{aligned} & f \in \left\{ Z^{\{1,\ldots,k\}}(s,0) \geq x + (c-m)(-s) \right\} \cap R \\ \Leftrightarrow \quad & f \in R, \quad -f_1(s) + \cdots - f_k(s) \geq x + (c-m)(-s) \\ \Leftrightarrow \quad & -\langle f, (\Gamma_1(s,\cdot), \ldots, \Gamma_k(s,\cdot)) \rangle_R \geq x + (c-m)(-s). \end{aligned}$$

Thus, the problem reduces to minimizing the Hilbert norm when the inner product with a fixed element is given, and the solution is a proper multiple of that element. It remains to minimize the norm of $((x + (c - m)t)/\sum v_i(t))(\Gamma_1(-t,\cdot), \ldots, \Gamma_k(-t,\cdot))$ with respect to $t > 0$.

Let $f^* \in R$ be a most probable path in a closed set $B \subset \Omega$ such that $f^* \neq 0$. We call the set

$$B^* \doteq \mathrm{cl}_{\,\Omega} \left\{ g \in R : \ \langle g - f^*, f^* \rangle_R \geq 0 \right\},$$

where $\mathrm{cl}_{\,\Omega} G$ denotes the closure of G in the topology of Ω, the *half-space approximation* of B. In particular, it is easy to see that

$$\{Q_0 \geq x\}^* = \{-Z_{-t^*} \geq x + (c-m)t^*\},$$

and the lower bound (8) is a consequence of the fact that in this case the half-space approximation is contained in the original set. See Figure 1.

It is worth of noting also that the most probable path vector in a set $\left\{ A_t^{\{1,\ldots,k\}} \geq y \right\}$, where $y > mt$, is in fact the conditional expectation

$$\mathbb{E}\left[(Z_s^{\{1\}}, \ldots, Z_s^{\{k\}}) \,\middle|\, A_t^{\{1,\ldots,k\}} = y \right].$$

This is a consequence of the fact that the conditional distribution of a Gaussian vector w.r.t. a linear condition is Gaussian, and its expectation equals the point where the density is highest.

More accurate estimates take, in some way or other, the geometry of the set $\{Q_0 \geq x\}$ around f^* into account. For different methods, see the books by Adler [4] and Piterbarg [25]. An original geometric reasoning, after transforming the problem into Fourier space, was given in [18].

Identifying most probable paths is interesting with its own rights — it is like "seeing what really happens" when the rare event occurs. For ordinary queues, this has mainly heuristic value, but we shall see that identifying these paths has an essential role in choosing a good approximation in the case of GPS and priority queues.

3.3 General Heuristic Approximations

Within logarithmic accuracy, the lower bound can be replaced by the still simpler approximate expression

$$\mathbb{P}(Q_t > x) \approx \exp\left(-\frac{(x + (c-m)u^*)^2}{2v(u^*)} \right), \tag{10}$$

which was called the *basic approximation* in [3]. Simulations of many cases indicate that the basic approximation may in fact be a general *upper bound* of $\mathbb{P}(Q_t > x)$, but no proof of this is known.

In all empirical studies and simulations one works in discrete time. The discrete time queue is always a little smaller than the corresponding continuous time queue. Indeed, if A is our continuous time model, the cumulative input process in discrete time is simply $(A_n)_{n\in\mathbb{Z}}$,

$$Q_n^{discr} = \sup_{m\leq n,\ m\in\mathbb{Z}} (A(m,n) - c(n-m)) \leq \sup_{s\leq n,\ s\in\mathbb{R}} (A(s,n) - c(n-s)) = Q_n^{cont}.$$

It was observed in [3] that one often gets fairly good approximations for a discrete time Gaussian queue Q^{discr} by multiplying the basic approximation by an appropriate constant p such that

$$p \lim_{x\to 0^+} \exp\left(-\frac{(x+(c-m)u_x^*)^2}{2v(u_x^*)}\right) \approx \mathbb{P}\big(Q_t^{discr} > 0\big)\,.$$

A good heuristic approximation for the non-emptiness probability of a discrete time queue with time resolution δ is (see [3])

$$\mathbb{P}\big(Q_t^{discr} > 0\big) \approx 2\mathbb{P}(A_\delta > c\delta)\,.$$

3.4 Approximations for GPS and Priority Queues

The structure of our method for getting estimates of queue length distributions in GPS and priority systems is the following. In order to get an approximation for $\left\{Q_0^{\{i\}} > x\right\}$, do

Step 1. Find the most probable path vector f^* of the event $\left\{Q_0^{\{1,\ldots,k\}} > x\right\}$. The path vector can be immediately written and plotted using Proposition 1.

Step 2. Check whether $Q_0^{\{1,\ldots,k\}\setminus\{i\}}(f^*) = 0$. If yes, go to Step 3, otherwise go to Step 4.

Step 3. (Empty Buffer Approximation) f^* is the most probable path vector in $\left\{Q_0^{\{i\}} > x\right\}$; use the corresponding half-space approximation. Stop.

Step 4. (Rough Full Link Approximation) Find a certain f^{RFLA}, where the only positive queue is $Q_0^{\{i\}}$ (or the others are much smaller); use the half-space approximation corresponding to f^{RFLA}.

The Empty Buffer Approximation uses the true most probable path vector, and it can be considered as reliable as the simple queue estimates of Section 3.2. In the Rough Full Link Approximation, the path vector f^{RFLA} also is just a heuristic approximation of the true most probable path vector. Both approximations are discussed in more detail below.

The Empty Buffer Approximation. The idea of the Empty Buffer Approximation (EBA), first studied by Berger and Whitt [7,8], is that in a two-class priority queue, the total queue usually consists almost exclusively of lower class traffic, and therefore its distribution is a good approximation to that of the pure lower class queue. Our approach gives a straightforward method to check the applicability of EBA in any particular combination of Gaussian traffic streams. Our examples indicate that EBA is a very good principle in most practically interesting priority scenarios with Gaussian traffic.

The EBA is also often useful in the study of a GPS system. However, it is never sufficient, because the classes are in a symmetric position in GPS, and the distribution of at most one class can be estimated with EBA.

Whereas it may require some work to check analytically whether the most probable path vector producing joint queue x satisfies the EBA condition, an approximately similar condition is much easier: in the two-class priority case, just check the "rough EBA condition"

$$-A^{\{1\}}_{-t^*}(f^*) \le ct^*. \tag{11}$$

This also leads to some interesting insight. Consider the priority system with two classes and assume, without restricting generality, that $m_1 = 0$. The condition 11 can be written as

$$\frac{x}{t^*} - m_2 \le \frac{v_2(t^*)}{v_1(t^*)}c. \tag{12}$$

In particular, we see that (12) holds if $m_2 \ge x/t^*$ (note, however, that t^* depends on the other quantities). In the special case that v_1 is a multiple of v_2, say $v_2(t) = av(t)$, $v_2(t) = bv(t)$, the condition becomes still simpler. Then t^* is independent of a and b, and we obtain the rather surprising result that when m_2 exceeds a certain threshold, then we are roughly in the EBA *irrespective* of the variance coefficients a and b! For example, if both $Z^{(i)}$'s are fractional Brownian motions with same self-similarity parameter H, then $t^* = Hx/((1-H)(c-m_2))$, which gives the condition $m_2 \ge (1-H)c$. The higher H, the lower is the threshold for m_2 above which a typical large class 2 queue consists of class 2 traffic alone.

The Rough Full Link Approximation. Consider the case of two traffic classes. For priority queues this does not restrict generality (since we neglect the effect of negative traffic). For GPS queues, the idea below could be extended to a larger number of classes, but the details would be much more complicated and, moreover, the heuristic probability estimates would be less reliable.

Consider a GPS system with weights μ_1 and μ_2. In the two class case, the priority system is obtained as the special case $\mu_2 = 0$. Assume that we are interested in the number $\mathbb{P}\left(Q^{\{2\}}_0 \ge x\right)$. As before, we first identify the most probable path pair f^* of $\left\{Q^{\{1,2\}}_0 \ge x\right\}$. If $Q^{\{1\}}_0(f^*) = 0$, we can use the EBA, as discussed in Section 3.4. So assume that $Q^{\{1\}}_0(f^*) > 0$.

The idea of our approximation in the non-EBA case is that any superfluous queue buildup decreases the likelihood of our path pair. Since we are only requiring that $Q^{\{2\}}_0(\omega)$ be big, $Q^{\{1\}}_0(\omega)$ must be close to zero with the optimal ω.

Thus, a class 2 queue of size x is most easily made so that the role of class 1 is essentially to fill its quota (in the priority case, to fill the whole link) without making a queue, while class 2 fills its quota and additionally builds a queue of size x.

To make this condition still simpler, we reduce this behavior to the one-dimensional conditions

$$\begin{aligned} A^{\{1\}}(-t,0) &= \mu_1 ct, \\ A^{\{2\}}(-t,0) &= \mu_2 ct + x, \end{aligned} \tag{13}$$

write down the most probable path pair fulfilling this, and finally minimize their norm with respect to t. We call this procedure the Rough Full Link Approximation (RFLA).

It is again an easy Hilbert space exercise, similar to Proposition 1, to determine the most probable paths in RFLA (see [15]):

Proposition 2. *The most probable path pair f^{RFLA} satisfying (13) is of the form*

$$\begin{aligned} f^{\mathrm{RFLA}}(\cdot) &= (f_1^{\mathrm{RFLA}}(\cdot), f_2^{\mathrm{RFLA}}(\cdot)) \\ &= \left(\frac{(\mu_1 c - m_1)t^*}{v_1(t^*)} \mathit{\Gamma}_1(t^*,\cdot),\ \frac{-x + (\mu_2 c - m_2)t^*}{v_2(t^*)} \mathit{\Gamma}_2(t^*,\cdot) \right), \end{aligned}$$

where $t^* < 0$ minimizes, w.r.t. t, the expression

$$\frac{(\mu_1 c - m_1)^2 t^2}{v_1(t)} + \frac{(x - (\mu_2 c - m_2)\, t)^2}{v_2(t)}. \tag{14}$$

In the case that both classes are Brownian motions (counterpart of Poisson processes), the RFLA gives the true most probable path pair in the non-EBA case. In general, however, the class 1 path in RFLA does not fill its quota over the whole interval $(-t^*, 0)$, thus part of class 2 traffic is "wasted", and there is a small class 1 queue at time 0, whereas the class 2 queue remains correspondingly smaller than x.

Using the reproducing kernel property and the fact that evaluation at a time point is a continuous linear functional both in R and $\mathit{\Omega}$, we see that the half-space corresponding to f^{RFLA} can be written as $E = \left\{Y \geq \|f^{\mathrm{RFLA}}\|_R^2\right\}$, where

$$Y = \frac{(\mu_1 c - m_1)t^*}{v_1(t^*)} Z_{t^*}^{\{1\}} + \frac{x - (\mu_2 c - m_2)t^*}{v_2(t^*)} Z_{t^*}^{\{2\}}.$$

Thus, our RFLA approximation, which the simulations indeed indicate to be a lower bound, is

$$\begin{aligned} \mathbb{P}\left(Q_0^{\{2\}} \geq x\right) &\approx \mathbb{P}(E) \qquad (15) \\ &= \overline{\mathit{\Phi}}\left(\sqrt{\frac{(\mu_1 c - m_1)^2\, t^{*2}}{v_1(t^*)} + \frac{(x - (\mu_2 c - m_2)\, t^*)^2}{v_2(t^*)}} \right). \end{aligned}$$

In order to check accuracy of our estimates, we have compared them to the empirical measures calculated from simulations. The simulation traces were generated using an extension of random midpoint displacement algorithm (RMD$_{mn}$, see [21]). Many examples are included in the papers [3,15], and they show reasonable accuracy of the method. In particular, the "basic approximations" turn always out to be upper bounds and the probabilities of the half-space approximations lower bounds. In the present overview paper, we restrict to the following example taken from [15].

Example: two fBm traffic classes with same self-similarity parameter. Let us consider a GPS system with two classes with $v_i(t) = \sigma_i^2 t^{2H}$ for $i = 1, 2$, and the parameter H is any number in $(0, 1)$. In this case we can compute the above quantities analytically.

First, fix $x > 0$ and consider the total queue. We have (see, e.g., [3])

$$t^* = \frac{Hx}{(1-H)(c-m)}.$$

Second, the rough EBA criterion (cf. (12)) for estimating class 1 reads

$$(\mu_2 c - m_2)t^* \geq f_2^*(t^*) = (x + (c-m)t^*)\frac{\sigma_2^2}{\sigma_1^2 + \sigma_2^2}.$$

Substituting t^*, we obtain the criterion

$$\frac{(\mu_2 c - m_2)H}{c-m} \geq \frac{\sigma_2^2}{\sigma_1^2 + \sigma_2^2}. \tag{16}$$

Note that only the mean and service rates appear on left and only the variance coefficients on right. If (16) is satisfied, the "basic approximation" reads

$$\mathbb{P}\left(Q_0^{\{1\}} \geq x\right) \approx \mathbb{P}\left(Q_0^{\{1,2\}} \geq x\right) \approx \exp\left(-\frac{(c-m)^{2H}}{\sigma_1^2 + \sigma_2^2} \cdot \frac{x^{2-2H}}{2\kappa(H)^2}\right),$$

where $\kappa(H) = H^H(1-H)^{1-H}$.

Third, if (16) does not hold, we use the RFLA. The squared R-norm of the most probable path in the set

$$\left\{-A_{-t}^{\{1\}} \geq \mu_1 ct + x,\ -A_{-t}^{\{2\}} \geq \mu_2 ct\right\}$$

is

$$\frac{((\mu_1 c - m_1)t + x)^2}{\sigma_1^2 t^{2H}} + \frac{(\mu_2 c - m_2)^2}{\sigma_2^2} t^{2-2H}.$$

The minimum is obtained at $t^* = \eta x$, where η is the positive root of a quadratic equation:

$$\eta = \frac{b + \sqrt{b^2 + 4aH}}{2a}, \text{ where}$$
$$a = (\frac{(\mu_1 c - m_1)^2}{\sigma_1^2} + \frac{(\mu_2 c - m_2)^2}{\sigma_2^2})(1-H),\ b = \frac{(\mu_1 c - m_1)(2H-1)}{\sigma_1^2}.$$

The basic approximation of $\mathbb{P}\left(Q_0^{\{1\}} \geq x\right)$ is then

$$\begin{aligned}&\mathbb{P}\left(Q_0^{\{1\}} \geq x\right) \qquad (17)\\ &\approx \exp\left(-\frac{1}{2}\left(\frac{((\mu_1 c - m_1)\eta + 1)^2}{\sigma_1^2 \eta^{2H}} + \frac{(\mu_2 c - m_2)^2}{\sigma_2^2}\eta^{2-2H}\right) x^{2-2H}\right).\end{aligned}$$

Simulations indicate that the approximations of this section work quite well — see [14,13,15].

4 "Phase Transitions" of Typical Queues

Even for simple queues, the most probable paths need not be unique. Nice examples of this were found by P. Mannersalo by superposing a periodic source and a fBm source [3,15]. A kind of phase transition was observed: typical small queues were caused by the periodical fluctuation of the periodic traffic, whereas typical long queues were caused by sustained heightened activity of the fBm traffic. (Cf. also [12].)

Another and, most importantly, non-artificial example was encountered by Pazhyannur and Fleming [24]. They studied a queue with input consisting of periodic coded voice traffic, modelled as follows. A source transmits with period d and uniformly distributed phase U. Volume in ith period is X_i. The X_i's can be strongly dependent. There are n i.i.d. sources. See Figure 2.

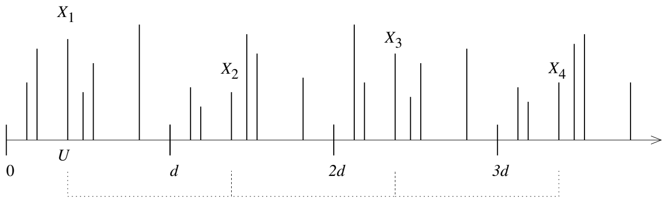


Fig. 2. The structure of vocoder traffic in [24]. Each source transmits periodically bursts whose sizes are random but correlated. The X_i's in the picture come from the same source.

Assuming that the number of sources is large enough for Gaussian modelling, our technique can be applied in a straightforward way. We only need to compute $v(t)$ for a single source — using a mathematical computer tool, the rest follows "according to the recipe". Denote the phase of our source by U and choose, for simplicity, $d = 1$. Then

$$\begin{aligned}A_t &= \sum_{i=1}^{\lfloor t \rfloor} X_i + 1_{\{U < t - \lfloor t \rfloor\}} X_{\lfloor t \rfloor + 1}\\ v(t) &= t \operatorname{Var}(X_0)\end{aligned}$$

$$+2\sum_{k=1}^{\lfloor t\rfloor}(t-k)\mathrm{Cov}\,(X_0,X_k)+(t-\lfloor t\rfloor)(1-(t-\lfloor t\rfloor))(\mathbb{E}\,\{X_0\})^2$$

$$\Gamma(s,t)=\frac{1}{2}(v(|s|)+v(|t|)-v(|s-t|)$$

Consider, as an example, the case $m=0$, $\mathrm{Cov}\,(X_0,X_k)=\rho^k$, $\rho=0.9$. Figure 3 shows a clear bend in the complementary distribution function of the queue length (resembling the shift from the "cell scale queue" to the "burst scale queue" in many ATM analyses — see, e.g., [22]). What happens when the queue size increases from 0.3 to 0.4?

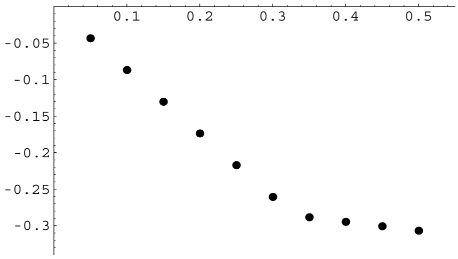


Fig. 3. Estimate of $\log_{10}\mathbb{P}(V>x)$.

Look first at the function $h(t)=(x+t)^2/v(t)$, which has to be minimized with respect to t. For $x=0.3$ or smaller, we have $t^*\approx x$, whereas for $x=0.4$, $t^*\approx 4$. Somewhere between 0.3 and 0.4 is a value of $x=x_0$ where the two local minima are equal. As a function of x, t^* makes big jump at x_0.

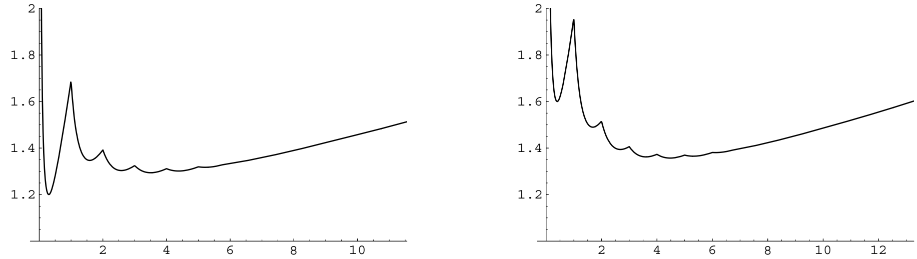


Fig. 4. Plot of the function $\dfrac{(x+t)^2}{v(t)}$. Left: $x=0.3$. Right: $x=0.4$.

Finally, the most probable paths shows that there is a very clear difference between typical queues of sizes 0.3 and 0.4. In the former, the queue is caused only by the bursts from different users, which are independent. In the latter, the busy period is larger than the period of the sources, which has the effect that the strong correlations between bursts of each source have become dominant, and the distribution tail decreases much slower than it did for small x's. Pazhyannur and Fleming discovered this queue behavior originally using more traditional heavy traffic approximations, but our method added an immediate visual insight which agreed with their interpretation. Moreover, they found that the Gaussian approximations were also quantitatively quite good.

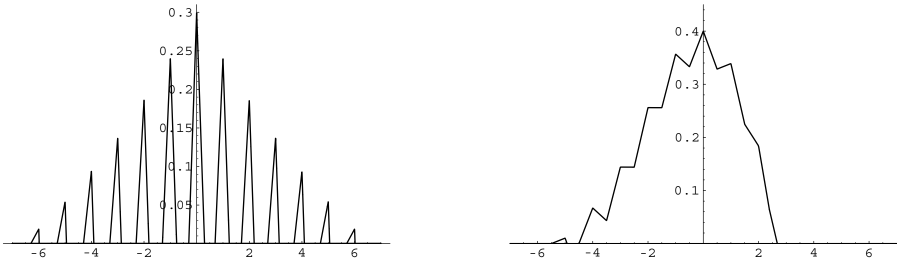


Fig. 5. Most probable queue path. Left: $x = 0.3$. Right: $x = 0.4$.

5 A Simple Model for Bandwidth Allocation by Prediction

Our last example analyses the performance of a queue whose service capacity is dynamically adjusted according to predicted demand, with a fixed prediction delay. The following setup is probably the simplest possible model for that kind of system.

Let A_t again be a Gaussian traffic process with parameters m and $v(t)$. Assume that instead of a fixed service rate, the service capacity is allocated dynamically with a delay $\varDelta$, with a relative surplus capacity ϵ. That is, we define the cumulative service process as

$$C_t \doteq (1+\epsilon)(A_{t-\varDelta} - A_{-\varDelta}) \tag{18}$$

(the last term is included in order to have $C_0 = 0$). The queue length process is

$$\begin{aligned} Q_t &= \sup_{s \le t}(A(s,t) - C(s,t)) \\ &\overset{\mathcal{D}}{=} \sup_{t \ge 0}(U_t - \epsilon m t), \end{aligned}$$

where $U_t = Z_t - (1+\epsilon)(Z_{t+\varDelta} - Z_{\varDelta})$. A straightforward computation gives

$$\mathrm{Var}\,(U_t) = (1 + (1+\epsilon)^2)v(t) - (1+\epsilon)(v(t-\varDelta) + v(t+\varDelta)) + 2(1+\epsilon)v(\varDelta).$$

In the space R we have

$$f(t) - (1+\epsilon)(f(t+\varDelta) - f(\varDelta)) = \langle f, \varGamma(t,\cdot) - (1+\epsilon)(\varGamma(t+\varDelta,\cdot) - \varGamma(\varDelta,\cdot))\rangle_R.$$

Thus, by our general method, the most probable path of Z creating a queue of size x at time 0 is

$$f_x^*(s) = -\frac{x + \epsilon m t^*}{\mathrm{Var}\,(U_{t^*})}(\varGamma(-t^*, s) - (1+\epsilon)(\varGamma(-t^*\varDelta, s) - \varGamma(-\varDelta, s)),$$

where $t = t^* > 0$ minimizes

$$\frac{(x+\epsilon m t)^2}{\mathrm{Var}\,(U_t)}.$$

In fact, the delay in such a system is bounded by $\varDelta$. The delay of a "fluid molecule" entering the system at time t can be expressed as

$$D_t \doteq \inf\left\{\tau :\ C(t, t+\tau) \geq Q_t\right\}.$$

Now,

$$\begin{aligned} Q_t - C(t,t+\tau) &= \sup_{s\leq t}(A(s,t) - (1+\epsilon)A(s-\varDelta, t-\varDelta)) \\ &\qquad -(1+\epsilon)A(t-\varDelta, t-\varDelta+\tau) \\ &= \sup_{s\leq t}(A(s,t) - (1+\epsilon)A(s-\varDelta, t-\varDelta+\tau)) \leq 0 \end{aligned}$$

for $\tau \geq \varDelta$, assuming that A_t is nondecreasing (which does not hold strictly for a Gaussian traffic model). (I thank P. Mannersalo for this insight.)

As an example, let us look at some paths in the case of fBm input $A_t = mt + \sigma Z_t$, where Z is a normalized fBm with self-similarity parameter H. The figures below were made with $\epsilon = 0.1$, $\varDelta = 1$, $m = 3$, $\sigma^2 = 1$, and $H = 0.75$.

Figure 6 compares the dynamically varied service with fixed service rate and same 10% overallocation. It is no surprise that very big queues arise when such a high load is offered to a fixed capacity server, whereas the queue remains essentially bounded in the former case (remember that the delays are strictly bounded). Figure 7 shows lower bound estimates of the complementary distribution functions. Indeed, the distribution tail of the dynamically served queue decreases very fast (faster than exponentially).

Figure 7 shows the most probable paths of the input rate and the queue of size 4. Note how cleverly our system makes its big (by its scale) queues: in order to fool the prediction, the input is first very slow and then, when the control cannot react any more, it suddenly speeds up. The queue path also has a noteworthy feature: after an input peak, the typical queue first decreases quickly, but then shifts to much slower decrease, whose slope corresponds to the overhead ϵ.

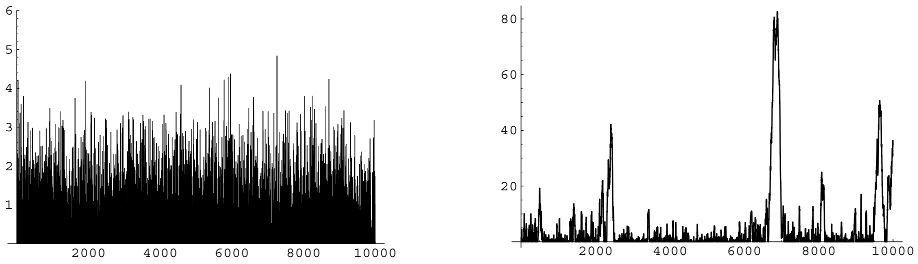


Fig. 6. Queue length processes of a system with prediction based dynamic allocation with $\epsilon = 0.1$, $\varDelta = 1$ (left), and a system with fixed service capacity $(1+\epsilon)m$ (right). The input processes are identical discrete time fBm traces with $m = 3$, $\sigma^2 = 1$, $H = 0.75$.

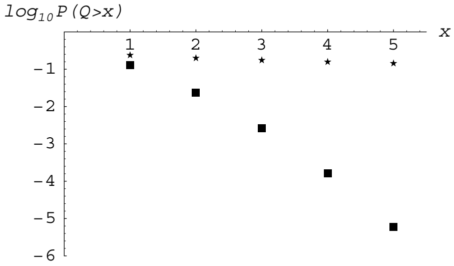


Fig. 7. Queue length distribution lower bounds $\log_{10} \ell(x)$ (see (8)) for a system with prediction based dynamic allocation with $\epsilon = 0.1$, $\varDelta = 1$ (squares), and a system with fixed service capacity $(1+\epsilon)m$ (stars). The input processes are fBm with $m = 3$, $\sigma^2 = 1$, $H = 0.75$.

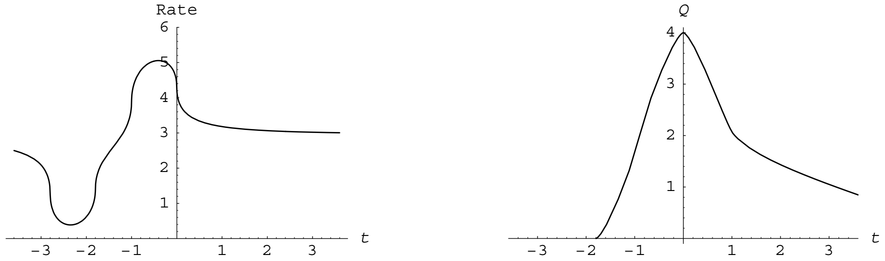


Fig. 8. The most probable path with queue size $x = 4$ in a system with prediction based dynamic allocation with $\epsilon = 0.1$, $\varDelta = 1$. The input process is fBm with $m = 3$, $\sigma^2 = 1$, $H = 0.75$. Left: rate. Right: queue length.

6 Conclusion

We have presented a straightforward method for studying various queueing systems with general Gaussian input traffic. These included priority queues, two-class GPS queues, and an example of dynamic server capacity allocation.

Using any advanced mathematical tool, it is possible to build expert systems, which make the analyses in this paper half or fully automatic once the parameters are given. In particular, the traffic in each class in described simply with mean rate and the cumulative variance function.

The novel theoretical aspect in this work is that we are looking for approximations and bounds in a Gaussian space — not large deviation theorems, which at least "officially" tell only about certain logarithmic limits. Although most of our quantitative estimates are more or less heuristic, we hope that this new point of view to queueing phenomena will prove fruitful in rigorous mathematics also. One of the key challenges may then be understanding the geometry of the threshold exceedance set in the neighborhood of the most probable path.

References

1. R.G. Addie. On weak convergence of long range dependent traffic processes. *Journal of Statistical Planning and Inference*, 80(1-2):155–171, 1999.
2. R.G. Addie, P. Mannersalo, and I. Norros. Performance formulae for queues with Gaussian input. In P. Key and D. Smith, editors, *Teletraffic Engineering in a Competitive World. Proceedings of the International Teletraffic Congress — ITC-16*, pages 1169–1178, Edinburgh, UK, 1999. Elsevier.
3. R.G. Addie, P. Mannersalo, and I. Norros. Most probable paths and performance formulae for buffers with Gaussian input traffic. To appear in European Transactions on Telecommunications, 2002.
4. R.J. Adler. *An Introduction to Continuity, Extrema, and Related Topics for General Gaussian Processes*, volume 12 of *Lecture Notes-Monograph Series.* Institute of Mathematical Statistics, 1990.
5. R. Azencott. *Ecole d'Eté de Probabiltés de Saint-Flour VII-1978*, chapter Grandes deviations et applications, pages 1–176. Number 774 in Lecture notes in Mathematics. Springer, Berlin, 1980.
6. R.R. Bahadur and S.L. Zabell. Large deviations of the sample mean in general vector spaces. *Ann. Prob.*, 7(4):587–621, 1979.
7. A.W. Berger and W. Whitt. Effective bandwidths with priorities. *IEEE/ACM Transactions on Networking*, 6(4), 1998.
8. A.W. Berger and W. Whitt. Extending the effective bandwidth concept to networks with priority classes. *IEEE Communications Magazine*, August 1998.
9. J.-D. Deuschel and D.W. Stroock. *Large Deviations.* Academic Press, Boston, 1989.
10. J. Kilpi and I. Norros. Testing the Gaussian character of access network traffic. Technical Report COST279TD(01)03, COST, 2001. Available from `http://www.vtt.fi/tte/projects/cost279/`.
11. W.E. Leland, M.S. Taqqu, W. Willinger, and D.V. Wilson. On the self-similar nature of Ethernet traffic (extended version). *IEEE/ACM Transactions on Networking*, 2(1):1–15, February 1994.

12. M.R.H. Mandjes and J.H. Kim. An analysis of the phase transition phenomenon in packet networks. To appear in Adv. or J. of Applied Probability.
13. P. Mannersalo and I. Norros. GPS schedulers and Gaussian traffic. Infocom 2002, New York.
14. P. Mannersalo and I. Norros. Gaussian priority queues. In *Proceedings of ITC 17*. Elsevier, 2001.
15. P. Mannersalo and I. Norros. A most probable path approach to queueing systems with general Gaussian input. *Computer Networks*, 2002. To appear.
16. L. Massoulie. Large deviations estimates for polling and weighted fair queueing service systems. *Adv. Perf. Anal.*, 2(2):103–127, 1999.
17. L. Massoulie and A. Simonian. Large buffer asymptotics for the queue with FBM input. *J. Appl. Prob.*, 36(3):894–906, 1999.
18. O. Narayan. Exact asymptotic queue length distribution for fractional Brownian traffic. *Advances in Performance Analysis*, 1:39–63, 1998.
19. I. Norros. A storage model with self-similar input. *Queueing Systems*, 16:387–396, 1994.
20. I. Norros. Busy periods of fractional Brownian storage: a large deviations approach. *Adv. Perf. Anal.*, 2:1–19, 1999.
21. I. Norros, P. Mannersalo, and J.L. Wang. Simulation of fractional Brownian motion with conditionalized random midpoint displacement. *Adv. Performance Anal.*, 2:77–101, 1999.
22. I. Norros, J.W. Roberts, A. Simonian, and J.T. Virtamo. The superposition of variable bit rate sources in an ATM multiplexer. *IEEE JSAC*, 9(3):378–387, April 1991.
23. A.K. Parekh and R.G. Gallager. A generalized processor sharing approach to flow control in integrated services network: the single node case. *IEEE/ACM Transaction on Networking*, 1(3):344–357, 1993.
24. R.S. Pazhyannur and P. Fleming. Asymptotic results for voice delay in packet networks. In *Vehicular Technology Conference / Fall*, 2001.
25. V.I. Piterbarg. *Asymptotic Methods in the Theory of Gaussian Processes and Fields.* American Mathematical Society, 1996.
26. P. Tran-Gia and N. Vicari, editors. *Impacts of new services on the architecture and performance of broadband networks. COST 257 Final Report.* compuTEAM Würzburg, 2000. `http://nero.informatik.uni-wuerzburg.de/cost/Final/`.

On the Queue Tail Asymptotics for General Multifractal Traffic

Sándor Molnár[1], Trang Dinh Dang[1], and István Maricza[1]

High Speed Networks Laboratory,
Dept. of Telecommunications and Telematics,
Budapest University of Technology and Economics
H–1117, Magyar tudósok körútja 2, Budapest, Hungary
Tel: (361) 463 3889, Fax: (361) 463 3107
{molnar, trang, maricza}@ttt-atm.ttt.bme.hu

Abstract. The tail asymptotics in an infinite capacity single server queue serviced at a constant rate and driven by general *multifractal* input process is presented. It has been shown that in the important subcase of the *monofractal* Fractional Brownian Motion (FBM) input traffic our result gives the well-known Weibullian tail. Practical engineering applications and validation of the results based on the analysis of measured network traffic have also been presented.

1 Introduction

Teletraffic research papers have reported the *high variability* and *burstiness* nature of network traffic in several LAN/WAN environments in the last decade. Moreover, it seems that most of the measured network traffic exhibits properties of *scale invariance*. It means that within a range of scales no characteristic dominant scale can be identified and some statistical properties within this range are not changing. This remarkable *scaling phenomenon* called for the *fractal modeling* of the investigated LAN/WAN traffic [21,20,9,19].

In the fractal modeling framework *long-range dependence* (LRD) and *self-similarity* have been analyzed intensively, and a number of studies is focused on how to detect accurately the LRD property and how to estimate the Hurst parameter [3,2]. LRD is revealed by the power law decay of the autocorrelation function at large lags, i.e., $r(k) \sim c|k|^{2H-2}, k \to \infty,\ H \in (0.5, 1)$, where c is a constant [3]. The degree of this slow decay is determined by the Hurst parameter (H).

A large group of traffic models (Fractional Brownian Motion (FBM) models, FARIMA models, Cox's M/G/∞ models, on/off models, etc.) to capture LRD and self-similar properties has also been developed [16]. Among these models the FBM [17] was found to be a popular parsimonious and tractable model of traffic aggregation [4,12]. The performance implications of the fractal property are also addressed in a series of studies [8,7].

After a number of new measurements and deeper analysis of network traffic it was discovered that the LAN/WAN traffic has a more complex scaling behaviour

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 105–116, 2002.

which cannot be described by LRD and self-similarity [21,9]. More precisely, it has been found that aggregate network traffic is asymptotically self-similar over time scales of the order of a few hundreds of milliseconds and above but it exhibits *multifractal* scaling below this time scale [9]. It has been also pointed out that the transition from the multifractal to self-similar scaling occurs around time scales of a typical packet round-trip time in the network [9]. However, some studies showed that multifractal scaling can also be present even at large time scales [15]. Therefore the monofractal traffic models (e.g. FBM) are inadequate to characterize the network traffic and multifractal traffic models with a much more flexible rule for the scaling law seem to be needed, especially for some WAN environments. Multifractal models can allow a compact description of a complex scaling behavior and it can also capture the non-Gaussian character of network traffic. Multifractal models imply the non-redundant scaling behavior of moments of many orders. The physical explanations and engineering implications are also addressed in several papers, e.g. [9].

A stochastic process $X(t)$ is called *multifractal* [13] if it has stationary increments and satisfies

$$\mathbb{E}[|X(t)|^q] = c(q)t^{\tau(q)+1} \tag{1}$$

for some positive q, where $\tau(q)$ is called the *scaling function* of multifractality and $c(q)$ is independent of t. An easy consequence of this definition is that $\tau(q)$ is a concave function [13]. If the scaling function $\tau(q)$ is a linear function of q the process is called *monofractal.* Multifractality is thus defined as a *global* property of the process moments. The definition is very general and it covers a very large class of processes. Multifractal processes are also called processes with *scaling property.*

From a practical point of view queueing analysis of fractal traffic is a very important issue for network dimensioning and management. Therefore the study of queueing systems with fractal traffic input is a challenge in queueing theory. In the recent years the performance of queues with LRD or self-similar input has been deeply analyzed. A collection of studies has proven that the FBM based models have a tail queue distribution that decays asymptotically like a Weibullian law, i.e., $\mathbf{P}[Q > b] \simeq \exp(-\delta b^{2-2H})$, where δ is a positive constant that depends on the service rate of the queue [17,6]. This important result shows that queues with FBM input ($H > 1/2$) have a much slower decay than that of the exponential.

However, there is a lack of queueing results available in the cases when the input traffic has a more complex scaling behaviour. Especially, queueing systems with multifractal input are an undiscovered field and only a few results have been published in the literature. Véhel *et al.* [22] suggested a cascade model for TCP traffic based on the retransmission and congestion avoidance mechanisms with no performance analysis. Riedi *et al.* [19,18] developed a multiscale queueing analysis in the case of tree-based multiscale input models. Gao *et al.* simulated queues fed by multiplicative multifractal processes in [10] but provided no analytical results. In contrast to these results we consider *general multifractal* *process* without any restrictions and derive *analytical results for the queue tail asymptotics.*

Our aim is to contribute to the queueing theory of multifractal queues and also to the traffic engineering implications. In this paper we present a novel analysis of multifractal queues including the tail asymptotics, special cases, and practical applications.

2 Queueing Model

We consider a simple queueing model: a single server queue in continuous time, the serving principle for offered work is defined to be FIFO (First In, First Out), the queue has infinite buffer and constant service rate s. Denote by $X(t)$ the total size of work arriving to the queue from time instant $-t$ in the past up to this moment, time instant 0. The so called *workload process* $W(t)$ is the total amount of work stored in the buffer in time interval $(-t, 0)$, i.e.,

$$W(t) = X(t) - st \tag{2}$$

Our interest, however, is the current buffer length of the queue, denoted by Q. This is the queue length in the equilibrium state of the queue when the system has been running for a long time and the initial queue length has no influence. If this state of the system does exist, i.e., stationarity and ergodicity of the workload process hold, and the stability condition for the system is also satisfied, i.e., $\limsup_t \mathbb{E}[X(t)]/t < s$, then:

$$Q = \sup_{t \geq 0} W(t), \tag{3}$$

where $W(0)$ is assumed to be 0. This equation is also referred to as *Lindley's equation.*

The input process $X(t)$ is considered as a general multifractal process which is defined by Eq. 1. This definition, presented by Mandelbrot *et al.* in [13], describes multifractal processes in terms of moments which leads to a more intuitive understanding of multifractality.

3 Approximation for Queue Tail Probabilities

We now state our main proposition:

Proposition 1. *The probabilities for the queue tail asymptotic of a single queueing model with general multifractal input is accurately approximated by:*

$$\log(\mathbf{P}[Q > b]) \approx \min_{q>0} \log \left\{ c(q) \frac{\left[\frac{b\,\tau_0(q)}{s(q-\tau_0(q))}\right]^{\tau_0(q)}}{\left[\frac{b\,q}{q-\tau_0(q)}\right]^{q}} \right\}, \quad b \text{ large} \tag{4}$$

where $\tau_0(q) := \tau(q) + 1$. The scaling function $\tau(q)$ and $c(q)$ are the functions which define the multifractal input process.

Proof
Using Lindley's equation the tail probabilities of queue length can be rewritten of the form: $\mathbf{P}[Q > b] = \mathbf{P}[\sup_{t\geq 0} W(t) > b]$. First let consider the quantity $\mathbf{P}[W(t) > b]$:

Replacing $W(t)$ by Eq. 2 we have

$$\begin{aligned} \mathbf{P}[W(t) > b] &= \mathbf{P}[X(t) - st > b] \\ &\leq \mathbf{P}[|X(t)| > b + st] \qquad (5) \\ &= \mathbf{P}[|X(t)|^q > (b+st)^q], \qquad \text{for arbitrary } q > 0 \\ &\leq \frac{E[X(t)^q]}{(b+st)^q}, \qquad \text{using Markov's inequality.} \qquad (6) \end{aligned}$$

Since the input process is multifractal defined by Eq. 1 then:

$$\mathbf{P}[W(t) > b] \leq \frac{c(q)t^{\tau_0(q)}}{(b+st)^q}$$

$$\Rightarrow \sup_{t\geq 0} \mathbf{P}[W(t) > b] \leq \sup_{t\geq 0} \frac{c(q)t^{\tau_0(q)}}{(b+st)^q} =: \sup_{t\geq 0} f(t). \qquad (7)$$

The straightforward derivation of $f(t)$ shows that it has a maximal value at $t = \frac{b\tau_0(q)}{s[q-\tau_0(q)]} > 0$. Therefore

$$\sup_{t\geq 0} \mathbf{P}[W(t) > b] \leq \sup_{t\geq 0} f(t) = c(q) \frac{\left[\frac{b\,\tau_0(q)}{s(q-\tau_0(q))}\right]^{\tau_0(q)}}{\left[\frac{b\,q}{q-\tau_0(q)}\right]^q}$$

$$\Rightarrow \log\left(\sup_{t\geq 0} \mathbf{P}[W(t) > b]\right) \leq \log\left(c(q) \frac{\left[\frac{b\,\tau_0(q)}{s(q-\tau_0(q))}\right]^{\tau_0(q)}}{\left[\frac{b\,q}{q-\tau_0(q)}\right]^q}\right), \qquad \text{for arbitrary } q > 0$$

$$\Rightarrow \log\left(\sup_{t\geq 0} \mathbf{P}[W(t) > b]\right) \leq \min_{q>0} \log\left(c(q) \frac{\left[\frac{b\,\tau_0(q)}{s(q-\tau_0(q))}\right]^{\tau_0(q)}}{\left[\frac{b\,q}{q-\tau_0(q)}\right]^q}\right). \qquad (8)$$

For a large class of stochastic processes (including FBM) the following limit holds [11]:

$$\lim_{b\to\infty} \frac{\log(\mathbf{P}[Q > b])}{\log(\sup_{t\geq 0} \mathbf{P}[W(t) > b])} = 1. \qquad (9)$$

In addition,

$$\log(\mathbf{P}[Q > b]) \geq \log(\sup_{t\geq 0} \mathbf{P}[W(t) > b]), \qquad (10)$$

then the right-hand side of Eq. 8 is a upper bound of a lower bound on $\log(\mathbf{P}[Q > b])$. The used inequalities in Eq. 10 and Eq. 6 become tight for finite large b.

Thus our approximation for the queue tail asymptotics is the following:

$$\log(\mathbf{P}[Q > b]) \approx \min_{q>0} \log \left(c(q) \frac{\left[\frac{b\,\tau_0(q)}{s(q-\tau_0(q))}\right]^{\tau_0(q)}}{\left[\frac{b\,q}{q-\tau_0(q)}\right]^{q}} \right), \qquad b \text{ large.}$$

□

For positive multifractal processes, i.e. $X(t) > 0$, Eq. 5 is an equality. In addition, the approximation in Eq. 10 and the inequality in Eq. 6 turn to be more accurate approximations as b tends to infinity. Thus the presented approximation is supposed to be asymptotically tight. The tightness and accuracy of the approximation is also experimentally investigated in Section V.

Considering the formula in Eq. 4 we see that it has an implicit form and just the given form of the functions $c(q)$ and $\tau(q)$ can provide the final result. The reason behind this is that the definition for the class of multifractal processes gives no restrictions for the functions $c(q)$ and $\tau(q)$ (beyond that $\tau(q)$ is concave). *Our conjecture is that the analysis of queueing systems with general multifractal input may produce some similar general results.* It means that there is no general queueing behaviour for these systems as the Weibullian decay in the case of Gaussian self-similar processes [17]. An actual multifractal model will determine, for example, the queue length probabilities of the system.

4 Applications

4.1 Fractional Brownian Motion

As a simple application first we consider a monofractal Gaussian process, called Fractional Brownian Motion (FBM). FBM is self-similar which is a simple case of monofractality and it is also Gaussian. The increment process of FBM is called Fractional Gaussian Noise (FGN). Queueing analysis of a single queue with FBM input is first presented by Norros [17] which showed the Weibullian decay for the asymptotic tail behaviour, i.e., $\mathbf{P}[X > x] \sim \exp(-\gamma x^{\beta})$ with $\beta \leq 1$. This result is also justified by Large Deviation techniques in [6]. Applying this input process model to our formula should show its use and robustness when comparing to these available results.

First we prove that any Gaussian process with scaling property is in the class of monofractal processes. Furthermore we give the explicit forms for $\tau(q)$ and $c(q)$.

Consider the following lemma:

Lemma 1. *A Gaussian process with scaling property is monofractal with parameters*

$$\begin{cases} \tau(q) = \frac{q}{2}\left[\tau(2)+1\right]-1 \\ c(q) \; = \frac{[2c(2)]^{q/2}}{\sqrt{\pi}} \Gamma\left(\frac{q+1}{2}\right), \end{cases}$$

where $\Gamma(\cdot)$ denotes the Gamma function, $\Gamma(z) = \int_0^{+\infty} x^{z-1} \exp^{-x} \mathrm{d}x,\ z > 0.$

The proof of this Lemma is provided in [5].

Turning back to our case of FBM with $c(2) = 1$ and $\tau(2) = 2H - 1$ where H is referred to as the Hurst parameter, we have

$$\begin{cases} \tau(q) = qH - 1 \\ c(q) = \frac{2^{q/2}}{\sqrt{\pi}} \Gamma\left(\frac{q+1}{2}\right). \end{cases}$$

Insert these two functions into our formula in Eq.4 we get

$$\log(\mathbf{P}[Q > b]) \approx \log\left(\min_{q>0} \left\{ \frac{2^{q/2}}{\sqrt{\pi}} \Gamma\left(\frac{q+1}{2}\right) \frac{\left(\frac{bH}{s(1-H)}\right)^{qH}}{\left(\frac{b}{1-H}\right)^{q}} \right\} \right) =: \log(\min_{q>o} g(q)).$$

The minimum value of the $g(q)$ for $q > 0$ function can be easily determined by taking its derivatives. The result is the following:

$$\log(\mathbf{P}[Q > b]) \approx \log(\min_{q>o} g(q)) = \log\left(\frac{1}{\sqrt{\pi}} \frac{\Gamma\left(\Psi^{-1}(\log K)\right)}{K^{\Psi^{-1}(\log K) - 1/2}} \right) =: \log(T_{FBM}(H, s, b)), \tag{11}$$

where $K = K(H, s, b) = \frac{1}{2} b^{2(1-H)} s^{2H} (1-H)^{-2(1-H)} H^{-2H}$, $\Psi(\cdot)$ is the *digamma* function, $\Psi(x) = \frac{\mathrm{d}}{\mathrm{d}x} \log \Gamma(x) = \frac{\Gamma'(x)}{\Gamma(x)}$, and $\Psi^{-1}(\cdot)$ denotes the inverse function of $\Psi(\cdot)$.

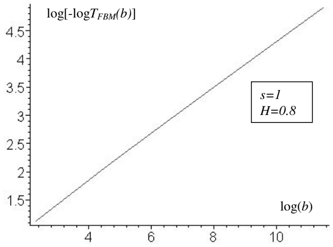


Fig. 1. By setting fixed values for H and s, the line in the log-log plot of $-\log T_{FBM}(b)$ versus b clearly shows the Weibullian decay for $T_{FBM}(H, s, b)$.

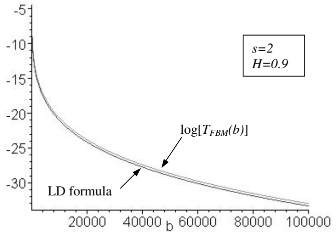


Fig. 2. Our approximation compared to the Large Deviation technique result.

The $T_{FBM}(H, s, b)$ function is quite complex with the presence of Gamma, digamma, and its inverse function. However, we have quite a good approximation of $T_{FBM}(H, s, b)$:

Proposition 2. *The approximation*

$$\frac{1}{\sqrt{\pi}} \frac{\Gamma\left(\Psi^{-1}(\log x)\right)}{x^{\Psi^{-1}(\log x) - 1/2}} \approx \exp(-x) \tag{12}$$

holds for large x, $x > 0$.

The proof and the precise sense of this approximation can be found in [5].

Applying this approximation we find that the queue tail for the FBM case satisfies:

$$\log\left(T_{FBM}(H,s,b)\right) \approx -\frac{1}{2}b^{2(1-H)}s^{2H}(1-H)^{-2(1-H)}H^{-2H}, \qquad b \text{ large.} \quad (13)$$

Eq. 13 shows the Weibullian decay of this queue which was first recognized and proven by Norros [17]. Numerical evaluations of the result are presented in Fig. 1 and Fig. 2. In Fig. 1 we fix the values of H and s and then calculate the values of the queue tail approximation $T_{FBM}(H,s,b)$ versus the queue size b and then plot it in the log-log scale. The linearity of the plot also demonstrates the Weibullian decay.

Now we compare our result to the result obtained by Duffield and O'Connell. The asymptotic formula for queue tail probabilities provided by Large Deviation technique presented in [6] is

$$\begin{aligned}\lim_{b\to\infty} b^{-2(1-H)}\log \mathbf{P}[Q>b] &= -\inf_{c>0} c^{-2(1-H)}\frac{(c+s)^2}{2}\\ \Leftrightarrow \log \mathbf{P}[Q>b] &\to -\frac{1}{2}b^{2(1-H)}s^{2H}(1-H)^{-2(1-H)}H^{-2H}, \quad \text{as } b\to\infty, \quad (14)\end{aligned}$$

where s also denotes the service rate. Therefore we can conclude that our approximation yields the Large Deviation result, see Eq. 13 and Eq. 14. The two results are depicted in Fig. 2 and we can see that the plots almost coincide for all calculated values of the queue size.

Our conclusions can be summarized in two main points: (i) the asymptotic tail approximation for the case of FBM has Weibullian decay; (ii) this result is also consistent with the formula presented by Norros [17] and by Duffield *et al.* with Large Deviation technique [6].

In the case of $H = 1/2$ (Brownian Motion) the above formula results in $\log \mathbf{P}[Q > b] \approx -2sb/\sigma^2$ where σ^2 denotes the variance of the process, which is in agreement with the queueing formula known from the theory of Gaussian processes [14,6].

4.2 Practical Solutions

We show here the practical use of the formula. Assume that we are interested in the behaviour of the tail of the steady-state buffer occupancy (queue length) distribution at a specific multiplexer in our network. The first step should be the fine resolution measurements of the input process. We also assume that the input process exhibits multifractal scaling properties. Then the scaling function $\tau(q)$ and the function $c(q)$ can be estimated from the collected data for some available parameters $q > 0$. *We emphasize the importance of the function $c(q)$ as the quantity factor of multifractal processes which is sometimes neglected in a number of studies dealing with multiscaling properties of the high-speed network traffic. The scaling function $\tau(q)$ defines only the quality of multiscaling and it is not enough for the description of a multifractal model and therefore for the analysis of queueing models with multifractal input processes.*

Now we suggest two practical methods for the approximation of the queue tail distribution:

1. Given the service rate s and the two sets $\{c(q)\}$ and $\{\tau(q)\}$, using Eq. 4 the approximation of $\log(\mathbf{P}[Q > b])$ can be computed for each value of b. This method is very simple but it is the more useful from network planning and capacity dimensioning point of view since we are only interested in some values of the tail probabilities. We mainly focus on the practical use of this method in this study.
2. The input process is fitted to a multifractal model. The two measured sets of $c(q)$ and $\tau(q)$ are fitted by $\tilde{c}(q)$ and $\tilde{\tau}(q)$. Then the analysis of the Eq. 4 with these functions can result in simple closed form of the queue tail probabilities. We use this method when studying the queue tail behaviour of a multifractal model.

5 Queueing Analysis

In this section we show the validation for the mentioned practical solution presented above by the queueing analysis of some real traffic traces. We also provide a simple method for estimation of multiscaling functions $c(q)$ and $\tau(q)$.

5.1 Simple Method for Multiscaling Functions Estimation

The full description of a multifractal model involves both $c(q)$ and the scaling function $\tau(q)$. We present here a simple method for testing of scaling properties and also for the estimation of these functions.

The definition of multifractal processes (Eq. 1) claims the stationarity condition for the increments. Therefore it is easy to verify the following relation for the moments of the increments: $\mathbb{E}[|Z^{(\triangle t)}|^q] = c(q)(\triangle t)^{\tau(q)+1} = c(q)(\triangle t)^{\tau_0(q)}, q > 0$, where $Z^{(\triangle t)}$ denotes the increment process of time sample $\triangle t$. Thus $\mathbb{E}[|Z^{(m\triangle t)}|^q]$ $= c(q)(m\triangle t)^{\tau_0(q)}, q > 0$ also holds for $m = 1, 2, \ldots$

Choose $\triangle t$ as the time unit, then

$$\log \mathbb{E}[|Z^{(m)}|^q] = \tau_0(q) \log m + \log c(q), \qquad q > 0. \tag{15}$$

Based on this property, the method is the following: Given a data series of a process increments $Z_1, Z_2, \ldots, Z_n$ and define its corresponding *real* aggregated sequence $\{Z^{(m)}\}$ of the aggregation level m by

$$Z_k^{(m)} = Z_{(k-1)m+1} + Z_{(k-1)m+2} + \ldots + Z_{km}, \qquad k, m = 1, 2, \ldots \tag{16}$$

If the sequence $\{Z_k\}$ has scaling property then the plot of absolute moments $\mathbb{E}[|Z^{(m)}|^q]$ versus m on a log-log plot should be a straight line due to Eq. 15. The slope of the straight line provides the estimate of $\tau_0(q)$ and the intercept is the value for $\log c(q)$. The illustration of the method can be seen in Fig. 3.

Note that we have no need to estimate $c(q)$ and $\tau_0(q)$ for all positive value of q, which is an impossible task. In fact, the largest value of q we should considered depends on the interested finite queue length of the involved queue length probability, see below.

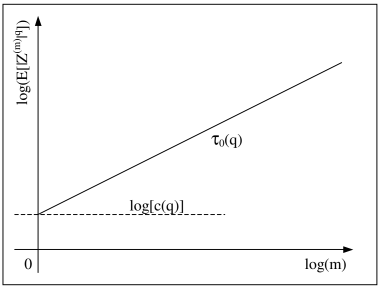


Fig. 3. A simple method for scaling test and the estimation of $c(q)$ and the scaling function $\tau(q)$.

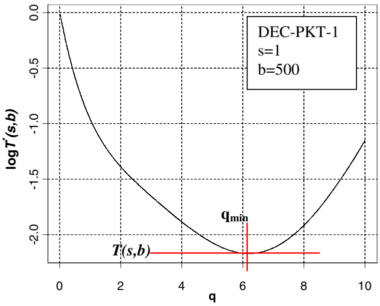


Fig. 4. Theoretical queue tail probability at each value of queue size b is the minimum of $\log T^*(s,b)$.

5.2 Analysis Results

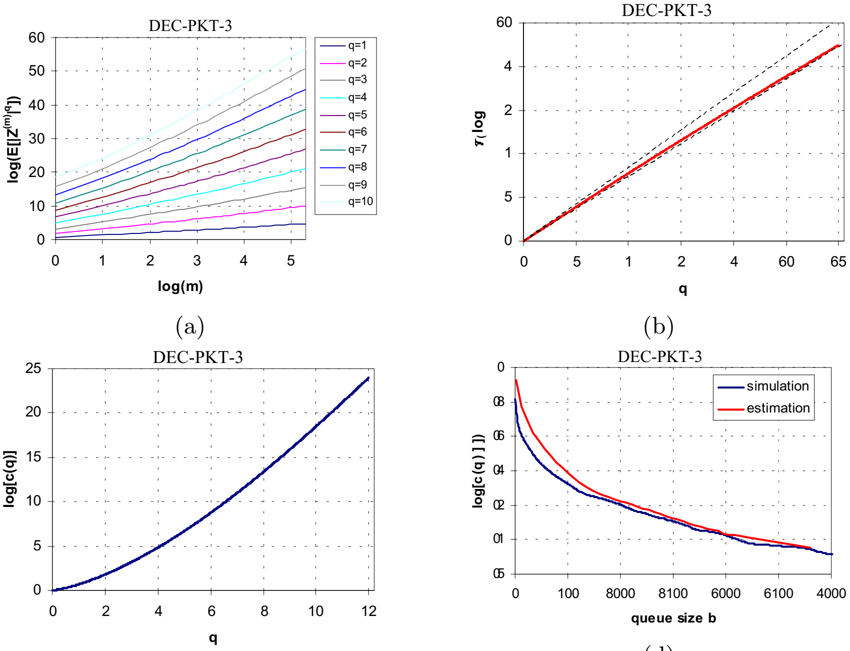
Our results have been first validated by simulation of multifractal cascades [5]. We have also carried out analysis of several measured IP packet arrival traffic traces (DEC-PKT-1, DEC-PKT-2, and DEC-PKT-3) obtained from the Internet Traffic Archive [1]. In this paper we present only two typical cases, i.e., monofractal (DEC-PKT-2) and multifractal traffic (DEC-PKT-3). The analysis validates the use of our approximation in a single queue with constant service rate and general multifractal input.

Figure 5(a) shows the plot of absolute moments of the aggregated sets of the set DEC-PKT-3 versus the aggregation level in a log-log plot for some values of moment q. The linearity of the plots observed in the figure clearly indicates the scaling property of this data set. After applying the estimation method we presented in the previous subsection we get the two sets of estimated $\tau_0(q)$ and $c(q)$ which are drawn in Fig. 5(b) and Fig. 5(c) (we estimate $\log c(q)$ instead of $c(q)$). The plot of the function $\tau_0(q) = \tau(q) + 1$ is a concave curve which suggests the multifractal property of DEC-PKT-3.

We then make a comparison between our approximation and the queueing simulation of real data traces to validate the use of the formula in practice. The approximation for probabilities of queue tail presented in Proposition 1 can be rewritten in the form

$$\begin{aligned}\log \mathbf{P}[Q > b] &\approx \min_{q>0}\left\{\log c(q) + \tau_0(q)\log\frac{b\tau_0(q)}{s(q-\tau_0(q))} - q\log\frac{bq}{q-\tau_0(q)}\right\}\\ &=: \min_{q>0}\{\log T^*(s,b)\} = T(s,b). \qquad (17)\end{aligned}$$

For the sake of calculation simplicity we choose the service rate such that $s = 1$. The lower curve in Fig. 5(d) shows the simulation result of the DEC-PFT-3 data set. Using Eq. 17 the value of the logarithmic tail probability at each concerned value of queue size b is taken by the numerical minimization of $\log T^*(s,b)$ with the estimated sets $\{c(q)\}$ and $\{\tau_0(q)\}$. An example is shown in Fig. 4.



(a) (b)

(c) (d)

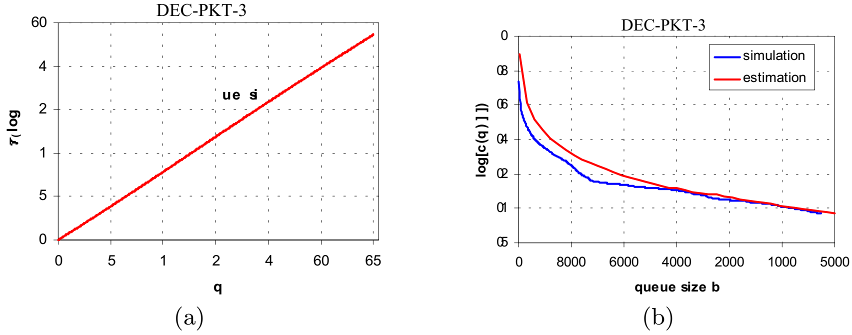
Fig. 5. Analysis results of the DEC-PKT-3 data set.

In addition, we do not need to plot $\log T^*(s,b)$ at each value of q to find its minimum. A simple program routine can do it for all concerned value of b at once. Our theoretical tail probabilities are on the upper curve in Fig. 5. As comparing with the simulation result which is seen in the same figure we found that it has the similar shape and becomes tight as b increases. This validates our result.

We have performed the same analysis with an other data set DEC-PFK-2. The results are summarized in Fig. 6. The DEC-PKT-2 data set, however, has the exact monofractal structure and can be well modelled by statistical self-similarity with Hurst parameter $H = 0.8$. Our queueing model deals with general multifractal input so it also involves the case of monofractal processes. Thus it is not surprising that the analysis also provides the correct queueing results in this case.

6 Conclusion

In this paper we studied the queueing performance of a single server infinite capacity queue with a constant service rate fed by general multifractal input process. We have provided the following results:



(a) (b)

Fig. 6. Analysis results of the DEC-PKT-2 data set.

(i) *We derived an asymptotic approximation of the steady-state queue length probabilities.*

(ii) *We showed that our results gives the well-known Weibullian queue tail in case of the monofractal Fractional Brownian Motion input process.*

(iii) *We proved that the class of Gaussian processes with scaling properties is limited to monofractal processes.*

(iv) *We demonstrated the practical applicability of our approximation and validated the method by queueing analysis of both multifractal and monofractal network traffic cases.*

There are several interesting topics for further research. Based on the multifractal process characterization one of our goal is to build a multifractal traffic model parameterized by the multifractal functions. We also intend to carry out more multifractal analyses of measured LAN/WAN traffic with corresponding performance analysis.

References

1. The internet traffic archive. http://ita.ee.lbl.gov.
2. P. Abry and D. Veitch. Wavelet analysis of long range dependent traffic. *IEEE Trans. Inform. Theory*, 44(1):2–15, January 1998.
3. J. Beran. *Statistics for Long-Memory Processes.* Chapman & Hall, One Penn Plaza, New York, NY 10119, 1995.
4. D. R. Cox. *Statistics: An Appraisal, Proc. 50th Anniversary Conference*, chapter Long-Range Dependence: A Review. Iowa State University Press, 1984.
5. T. D. Dang and S. Molnár. Queue asymptotics with general multifractal input. Technical report, Budapest University of Technology and Economics, July 2001.
6. N.G. Duffield and N. O'Connell. Large deviations and overflow probabilities for the general single-server queue, with applications. In *Proc., Cam. Phil. Soc.*, volume 118, pages 363–374, 1994.
7. A. Erramilli, O. Narayan, A. L. Neidhardt, and I. Saniee. Performance impacts of multi-scaling in wide-area TCP/IP traffic. In *Proc., IEEE INFOCOM 2000*, volume 1, pages 352–359, Tel Aviv, Israel, 2000.

8. A. Erramilli, O. Narayan, and W. Willinger. Experimental queueing analysis with long-range dependent packet traffic. *IEEE/ACM Trans. on Networking*, 4(2):209–223, April 1996.
9. A. Feldmann, A. C. Gilbert, and W. Willinger. Data Networks as Cascades: Investigating the Multifractal Nature of Internet WAN Traffic. *ACM Computer Communication Review*, 28:42–55, September 1998.
10. J. Gao and I. Rubin. Multifractal modeling of counting processes of long-range dependent network traffic. In *Proceedings SCS Advanced Simulation Technologies Conf.*, San Diego, CA, April 1999.
11. J. Hüsler and V. Piterbarg. Extremes of a certain class of Gaussian processes. *Stochastic Process. Appl.*, 83:257–271, 1999.
12. T. G. Kurtz. *Stochastic Networks: Theory and Applications*, chapter Limit Theorems for Workload Input Models. Oxford University Press, 1996.
13. B. B. Mandelbrot, A. Fisher, and L. Calvet. *A Multifractal Model of Asset Return.* Yale University, 1997. Working Paper.
14. M. B. Marcus and L. A. Shepp. Sample behaviour of Gaussian processes. In *Proceedings of the Sixth Berkeley Symposium*, 1972.
15. S. Molnár and T. D. Dang. Scaling analysis of IP traffic components. In *ITC Specialist Seminar on IP Traffic Measurement, Modeling and Management*, Monterey, CA, USA, 18-20 September 2000.
16. S. Molnár and A. Vidács. On Modeling and Shaping Self-Similar ATM Traffic. In *15th International Teletraffic Congress*, Washington, DC, USA, June 1997.
17. I. Norros. A storage model with self-similar input. *Queueing Systems*, 16:387–396, 1994.
18. V. J. Ribeiro, R. H. Riedi, M. S. Crouse, and R. G. Baraniuk. Multiscale queuing analysis of long-range-dependent network traffic. In *Proceedings of IEEE INFOCOM 2000*, Tel Aviv, Israel, March 2000.
19. R. H. Riedi, M. S. Crouse, V. J. Ribeiro, and R. G. Baraniuk. A multifractal wavelet model with application to network traffic. *IEEE Trans. Inform. Theory*, 45(3):992–1018, April 1999.
20. R. H. Riedi and J. Lévy Véhel. Multifractal properties of TCP traffic: a numerical study. INRIA research report 3129, Rice University, February 1997.
21. M. S. Taqqu, V. Teverovsky, and W. Willinger. Is network traffic self-similar or multifractal? *Fractals*, 5:63–73, 1997.
22. J. Lévi Véhel and B. Sikdar. A multiplicative multifractal model for TCP traffic. In *Proc., IEEE ISCC*, Hammamet, Tunisia, July 2001.

Some Models for Contention Resolution in Cable Networks

Onno Boxma[1], Dee Denteneer[2], and Jacques Resing[1]

[1] EURANDOM and Department of Mathematics and Computer Science
Eindhoven University of Technology
P.O. Box 513, 5600 MB Eindhoven, The Netherlands
{Boxma,Resing}@win.tue.nl

[2] Philips Research, Digital Signal Processing Group
Prof. Holstlaan 4, 5656 AA Eindhoven, The Netherlands
Dee.Denteneer@philips.com

Abstract. In this paper we consider some models for contention resolution in cable networks, in case the contention pertains to requests and is carried out by means of contention trees. More specifically, we study a number of variants of the standard machine repair model, that differ in the service order at the repair facility. Considered service orders are First Come First Served, Random Order of Service, and Gated Random Order of Service. For these variants, we study the sojourn time at the repair facility. In the case of the free access protocol for contention trees, the first two moments of the access delay in contention are accurately represented by those of the sojourn time at the repair facility under Random Order of Service. In the case of the blocked access protocol, Gated Random Order of Service is shown to be more appropriate.

1 Introduction

Cable networks are currently being upgraded to support bidirectional data transport, see e.g. van Driel *et al.* [1]. The system is thus extended with an 'upstream' channel to complement the 'downstream' channel that is already present. This upstream channel is time slotted and shared among many stations so that contention resolution is essential for upstream data transport. An efficient way to carry out the upstream data transport is via a request-grant mechanism, like in Digital Video Broadcasting [2]: stations request data slots in contention with other stations via contention trees. After a successful request, data transfer follows in reserved slots, not in contention with other stations.

A tractable model for the access delay due to this request procedure is an essential step toward a better understanding of such a request-grant mechanism, and expressions for the first moments of the distribution of the access delay are particularly relevant. However, the performance analysis of contention trees, see Mathys and Flajolet [3] or Tsybakov [4], has been carried out under the assumption of a Poisson source model. This does not easily lead to properties

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 117–128, 2002.

of the closed model for a finite number of stations that is appropriate when contention trees are used for reservation.

The machine repair model, also known as the computer terminal model or as the time sharing system (e.g. Kleinrock [5], Section 4.11; Bertsekas and Gallager [6], Example 3.22), is one of the key performance models that assumes a finite input population. Therefore, it is a promising model for contention resolution using contention trees. The basic model is as follows. There are N machines working in parallel. After a working period a machine breaks down and joins the repair queue. At the repair facility, a single repairman repairs the machines according to some service discipline. Once repaired, a machine starts working again. In the basic model, the distribution of both the working time and the repair time of machines is assumed to be exponential and the service discipline at the repair facility is assumed to be First Come First Served (FCFS).

In this paper, we show that the machine repair model can be an appropriate model for contention resolution in cable networks for the case that so-called Capetanakis-Tsybakov contention trees are used for reservation (see [7,8]). It turns out that the average time spent in contention resolution, obtained via simulations, matches the average sojourn time at the repair facility in the basic machine repair model almost perfectly. However, the basic model fails to accurately predict the *variance* of the time spent in contention resolution.

Closer inspection of contention trees reveals a possible source for this mismatch. Contention trees, to be described in Section 2, deviate from queues with a FCFS discipline in that each station in a given group has the same probability of being served, irrespective of the instant at which it entered the group. This suggests that variants of the basic machine repair model are needed to obtain a more appropriate model for the time spent in contention resolution, and that these variants should have some randomness built into their service discipline. In this paper, we consider two such variants.

Firstly, we consider the machine repair model as described above with a *random order of service* (ROS) discipline. Here, after a repair, the next machine to be repaired is chosen randomly from the machines in the repair queue. We analyse the sojourn time distribution at the repair queue for this model by exploiting a close relationship with the machine repair model considered in Mitra [9], in which the service discipline at the repair facility is *processor sharing* (PS).

We shall see that the variance of the sojourn time under ROS gives an accurate prediction of the access delay of requests in contention, when the so-called *free access protocol* is used. However, the prediction is not accurate in case of the so-called *blocked access protocol*. For that protocol, we consider an extension of the machine repair model. In this extension, machines that broke down are first gathered in a waiting room before they are put in random order in the actual repair queue at the instants that this repair queue becomes empty. In the sequel this service discipline will be called *gated random order of service* (GROS). For the GROS service discipline, just as for the ROS discipline, the average sojourn time at the repair facility is identical to the average sojourn time in case of the FCFS discipline – which, as mentioned above, accurately matches the mean time spent in contention resolution. Hence, the emphasis of our analysis will be on obtaining an (approximate) expression for the *variance* of the sojourn time at the repair facility.

It is appropriate to comment briefly on the relevance of the *variance* of the access delay in contention resolution. Firstly, low variability implies low jitter. As such, access variability is a key performance measure in itself. However, the main reason for studying the variance of the access delay is that it is needed in understanding the total *average* waiting time in cable networks. This follows from the request grant mechanism employed, as explained in the first paragraph of this introduction. Data transfer in cable networks consists of two stages. In the first stage, bandwidth for data transfer is being requested via the contention procedure. Once successfully transmitted, the requests queue up in a second queue. In this queue, the service time distribution is given by the distribution of the number of packets for which transfer is being requested. Now, due to the phenomenon of request merging, which will be described in more detail in Section 2, the number of packets being requested depends on the time spent in contention so that the variance of the service time depends on the variance of the access delay in the contention resolution. Clearly, the variance of the service time is needed to estimate the average waiting time in this second queue. In the present paper we concentrate on the first stage; to analyze the overall delay is a topic for further study.

The rest of the paper is organised as follows. In Section 2 we describe the contention resolution process using contention trees in more detail. In Section 3 we review some of the properties of the basic machine repair model. Moreover, we derive expressions for the first two moments of the steady state sojourn time distribution. The machine repair model with ROS service discipline is considered in Section 4. Here, we first relate the model with ROS service discipline to the model with PS. After that, we briefly review the main results from Mitra [9] for the model with PS. In Section 5, we give an approximate derivation of the moments of the sojourn time in the model with GROS service discipline. In Section 6 we present numerical results which show that the models of Section 4 and 5 can be used to approximate the sojourn time for contention resolution in cable networks using contention trees.

2 Access via Contention Trees

Tree algorithms are a popular tool to provide access to a channel that is time slotted and shared among many stations. These algorithms and their many variants are also referred to as stack algorithms or splitting algorithms; we refer to Bertsekas and Gallager [6], Section 4.3, for a survey. In this paper, we will confine attention to the basic ternary tree, illustrated in Figure 1. The basic tree consists of nodes, and each of these nodes comprises three slots of the access channel. A collision occurs if more than one station attempts a transmission in a slot. These collisions are then resolved by recursively splitting the set of colliding stations, plus possible newcomers as explained below, into three disjoint subgroups. For

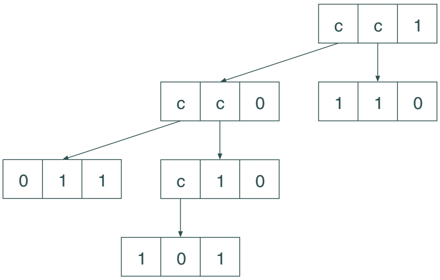


Fig. 1. Basic tree algorithm: slots of the tree with a collision (c) are recursively split until all slots are empty (0) or have a successful transmission (1)

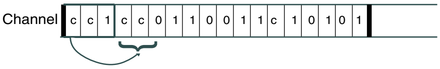


Fig. 2. Same tree as in Figure 1, with a breadth first ordering of the nodes

this, usually, a random mechanism is employed. This splitting continues until all tree slots are either empty or contain a successful transmission. This splitting process can be thought of as a tree, but takes place in time slots of the communication channel devoted to the contention resolution, so that the nodes of the tree must be time ordered. For this, we will use the breadth first ordering, as illustrated in Figure 2.

This basic tree algorithm must be complemented with a 'channel access protocol' that describes the procedure to be followed by stations that have data to transmit and that are not already contending in the tree. We consider two such access protocols: free access and blocked access. In the former protocol, access to the tree is free and any station can transmit a request in the next node of the tree, as soon as it has data to transmit. In the latter protocol, the tree is blocked so that new stations can only transmit requests in the root node of the tree that is started as soon as the current tree has been completed.

The stations exhibit the following behaviour:

- A station becomes active in the contention process upon generation of a data packet. In case of free access it will then transmit a request in the next tree node, randomly choosing one of the three slots in this node. In case of blocked access it will wait for the next new tree to be started and transmit its request in one of the slots of the root node of this tree.
- A station stays active until its request has been successfully transmitted.
- While active, the station can update its request (*request merging*). Hence, packets generated at such an active station do not cause extra requests.
- After successful transmission of the request, the station becomes inactive, to become active again upon the generation of a new data packet.

Note that request merging implies that the number of stations that can be active in contention is bounded. Exactly this property makes results on the performance of contention trees in open models, as investigated in e.g. Mathys and Flajolet [3] or Tsybakov [4], less relevant to contention resolution in cable networks. This property also explains the approach in this paper, which approximates the access delay in transmitting a request by means of the sojourn time in a machine repair model.

3 Properties of the Basic Machine Repair Model

First we introduce some notation and quote some properties of the basic machine repair model. The total number of machines in the system is denoted by N. The machines work in parallel and break down, independently, after an exponentially distributed working period with parameter λ. Machines that broke down queue up in the repair queue, where they are served FCFS by a single repairman. The repair times of machines are exponentially distributed with parameter μ.

With the random variables X and Y we denote the steady state number of machines that are in Q_W (i.e., are working) and that are in Q_R (i.e., are in repair), respectively. Clearly, the number of working machines and the number of machines in repair evolve as Markov processes. Their steady state distributions are (in fact even for generally distributed working periods, cf. [5,10]):

$$\Pr(X = k) = \Pr(Y = N - k) = \frac{\rho^k/k!}{\sum_{i=0}^{N} \rho^i/i!}, \qquad k = 0, \ldots, N, \tag{1}$$

where $\rho := \mu/\lambda$. For the mean and variance of X and Y we have

$$\mathrm{E}(X) = \rho(1 - B_N(\rho)), \quad \mathrm{E}(Y) = N - \mathrm{E}(X), \tag{2}$$

$$\mathrm{var}(X) = \mathrm{var}(Y) = \mathrm{E}(X) - \rho B_N(\rho)[N - \mathrm{E}(X)], \tag{3}$$

where $B_N(\rho)$ denotes Erlang's loss probability, which is given by

$$B_N(\rho) = \frac{\rho^N/N!}{\sum_{i=0}^{N} \rho^i/i!}. \tag{4}$$

Indeed, it is well known that the number of operative machines has the same distribution as the number of busy lines in the classical Erlang loss model.

We now turn to the moments of the sojourn time of an arbitrary machine at the repair facility. To this end, consider the time epoch at which an arbitrary machine breaks down and jumps to the repair queue. Stochastic quantities related to this instant will be denoted by a subscript 1. Thus X_1 is the number of working machines at this instant, and Y_1 is the number of machines in repair at this instant. From the arrival theorem, see Sevcik and Mitrani [11], it follows that the distributions of X_1 and Y_1 are given by (1), with N replaced by $N - 1$:

$$\Pr(X_1 = k) = \Pr(Y_1 = N - 1 - k) = \frac{\rho^k/k!}{\sum_{i=0}^{N-1} \rho^i/i!}, \qquad k = 0, \ldots, N - 1. \tag{5}$$

The sojourn time of an arbitrary machine at the repair facility equals its own repair time plus the sum of the repair times of the machines already present at the repair facility. Thus, denoting this sojourn time by S, we have that

$$S = \sum_{i=1}^{Y_1+1} B_i, \tag{6}$$

with $B_i, i = 1, 2, \ldots$, a sequence of independent, exponentially distributed random variables with parameter μ. Equation (6) enables us to obtain the Laplace-Stieltjes transform (LST) of the sojourn time at the repair facility (see also [10]). Here, however, we are mainly interested in the first two moments of the sojourn time. These can be obtained by consideration of the moments of the random sum, i.e.,

$$\mathrm{E}(S_{FCFS}) = \frac{1}{\mu}(N - \rho(1 - B_{N-1}(\rho))), \tag{7}$$

$$\mathrm{var}(S_{FCFS}) = \frac{1}{\mu^2}\left(N - \rho B_{N-1}(\rho)[N - 1 - \rho(1 - B_{N-1}(\rho))]\right). \tag{8}$$

Now, for N large and $N >> \mu/\lambda$, $B_N(\rho)$ goes to zero like $\rho^N/N!$. Hence, in that case, the following are extremely sharp approximations:

$$\mathrm{E}(S_{FCFS}) \approx \frac{N}{\mu} - \frac{1}{\lambda}, \quad \mathrm{var}(S_{FCFS}) \approx \frac{N}{\mu^2}. \tag{9}$$

In Sections 4 and 5 we shall study the sojourn time distribution at Q_R under the assumption that the service discipline at that queue is Random Order of Service (ROS) and Gated Random Order of Service (GROS), respectively. The *mean* sojourn time in Q_R is the same under FCFS, ROS and GROS; this is a direct consequence of Little's formula and the fact that the distribution of the number of customers in Q_R is the same for any work-conserving service discipline that does not pay attention to the actual service requests of customers. We therefore focus in particular on the *variance* of the sojourn time in Q_R. Formula (9) shows that for the FCFS discipline, asymptotically, this variance is linear in the number of machines and does not depend on λ, the parameter of the distribution of the working times.

4 The Model with ROS Service Discipline

Again we consider the basic machine repair model, but now the service discipline at Q_R is *random order of service*. For reasons that will soon become clear, we assume that the system contains not N but $N + 1$ machines. The main goals of this section are: (i) to determine the LST of the waiting time distribution at Q_R, (ii) to relate this distribution to the sojourn time distribution at Q_R in case the service discipline is PS instead of ROS, and (iii) to determine the asymptotic behaviour of the variance of the waiting (and sojourn) time at Q_R under the ROS discipline.

Consider a tagged machine, C, at the instant it arrives at Q_R. Let S_{ROS} (W_{ROS}) denote the steady state sojourn (waiting) time of C at Q_R. S_{ROS} is the sum of W_{ROS} and a service time that is independent of W_{ROS}, and hence we can concentrate on W_{ROS}. We denote by $Y_1^{(N+1)}$ the number of machines in Q_R, as seen by C upon arrival in Q_R. Introduce

$$\phi_j(\omega) := \mathrm{E}[\mathrm{e}^{-\omega W_{ROS}} | Y_1^{(N+1)} = j+1], \quad \mathrm{Re}\ \omega \geq 0, \quad j = 0, \dots, N-1.$$

We can write, for Re $\omega \geq 0$,

$$\mathrm{E}[\mathrm{e}^{-\omega W_{ROS}} | W_{ROS} > 0] = \sum_{j=0}^{N-1} \mathrm{P}(Y_1^{(N+1)} = j+1 | Y_1^{(N+1)} > 0) \phi_j(\omega). \qquad (10)$$

The following set of N equations for the N unknown functions $\phi_0(\omega), \dots, \phi_{N-1}(\omega)$ holds:

$$\phi_j(\omega) = \frac{\mu + (N-j-1)\lambda}{\mu + (N-j-1)\lambda + \omega} [\frac{(N-j-1)\lambda}{\mu + (N-j-1)\lambda} \phi_{j+1}(\omega) + \frac{\mu}{\mu + (N-j-1)\lambda} (\frac{1}{j+1} + \frac{j}{j+1} \phi_{j-1}(\omega))]. \qquad (11)$$

Notice that the pre-factors of $\phi_{-1}(\omega)$ and $\phi_N(\omega)$ equal zero. Formula (11) can be understood in the following way. The pre-factor $(\mu + (N-j-1)\lambda)/(\mu + (N-j-1)\lambda + \omega)$ is the LST of the time until the first 'event': Either an arrival at Q_R or a departure from Q_R. An arrival occurs first with probability $(N-j-1)\lambda/(\mu + (N-j-1)\lambda)$. In this event, the memoryless property of the exponential working and repair times implies that the tagged machine C sees the system as if it only now arrives at Q_R, meeting $j+2$ other machines there. A departure occurs first with probability $\mu/(\mu + (N-j-1)\lambda)$. In this event, C is with probability $1/(j+1)$ the one to leave the waiting room and enter the service position; if it does not leave, it sees Q_R as if it only now arrives, meeting j other machines there.

We can use (11) to obtain numerical values of $\mathrm{E}(W_{ROS}|W_{ROS} > 0)$ and $\mathrm{var}(W_{ROS}|W_{ROS} > 0)$. Formula (11) can also be used to study this mean and variance asymptotically, for $N \to \infty$. In fact, for this purpose we can also use the analysis given by Mitra [9] for a strongly related model: The machine-repair model with *processor sharing* at Q_R and with N (instead of $N+1$) machines. Denote the LST of the *sojourn* time distribution of a machine meeting j machines at Q_R, in the case of processor sharing, by $\psi_j(\omega)$. A careful study of Formula (11) and the explanation following it reveals that *exactly* the same set of equations holds for $\psi_j(\omega)$, if in the PS case there are not $N+1$ but N machines in the system. Not only do we have $\phi_j(\omega) = \psi_j(\omega)$, $j = 0, \dots, N-1$, but it also follows from (5) that $\mathrm{P}(Y_1^{(N+1)} = j+1 | Y_1^{(N+1)} > 0) = \mathrm{P}(Y_1^{(N)} = j)$, $j = 0, \dots, N-1$. The above equalities, combined with (10), imply that W_{ROS}, conditionally upon it being positive, in the machine-repair system with $N+1$ machines, has the same distribution as the sojourn time under processor sharing in the corresponding system with N machines. Adding a superscript (N) for the case of a machine-repair system with N machines, we can write:

$$\mathrm{P}(S_{PS}^{(N)} > t) = \mathrm{P}(W_{ROS}^{(N+1)} > t | W_{ROS}^{(N+1)} > 0). \qquad (12)$$

This equivalence result between ROS and PS may be viewed as a special case of a more general result in [12] (see [13] for another special case).

Using (12), it is easily verified that, for the machine repair model with N machines, $\mathrm{E}S_{ROS} = \mathrm{E}S_{PS} = \mathrm{E}S_{FCFS}$, just as indicated in Section 3. Multiplication by t and integration over t in (12) yields:

$$\mathrm{var}(S_{PS}^{(N)}) = \frac{\mathrm{var}(W_{ROS}^{(N+1)})}{\mathrm{P}(W_{ROS}^{(N+1)} > 0)}, \tag{13}$$

where $\mathrm{P}(W_{ROS}^{(N+1)} > 0)$ is easily obtained from (1). If N is large and $N > \mu/\lambda$, then $\mathrm{P}(W_{ROS}^{(N+1)} = 0)$ is negligibly small. The previous formula hence implies: For $N \to \infty$, $\mathrm{var}(S_{PS}^{(N)}) \sim \mathrm{var}(W_{ROS}^{(N)})$ – and hence also $\mathrm{var}(S_{PS}^{(N)}) \sim \mathrm{var}(S_{ROS}^{(N)})$.

For an asymptotic analysis of $\mathrm{E}W_{ROS}^{(N)}$ and $\mathrm{var}(W_{ROS}^{(N)})$ we can thus immediately apply corresponding asymptotics of Mitra [9] for the PS-variant. Mitra [9] shows that S_{PS} is hyper-exponentially distributed. This, in turn, immediately implies that (see Proposition 12 in [9]),

$$\mathrm{var}(S_{PS}) \geq (\mathrm{E}S_{PS})^2. \tag{14}$$

Hence $\mathrm{var}(S_{PS}) = \mathrm{O}(N^2)$ for $N \to \infty$, which sharply contrasts with the $\mathrm{O}(N)$ behavior for FCFS (cf. (9)).

5 The Model with GROS Service Discipline

In this section, we consider the model with GROS service discipline as described in Section 1. Again, we let Y denote the number of machines in the total waiting area (i.e. waiting room plus waiting queue). Obviously the distribution of Y equals the distribution of the number of machines in the repair queue in the standard model described in Section 3, and is given by (1).

We will now consider the sojourn time until repair, S_{GROS}, of an arbitrary (tagged) machine for the model with GROS service discipline. Observe that

$$S_{GROS} = \sum_{i=1}^{Y_1^{(1)}} B_i^{(1)} + \sum_{i=1}^{Y_1^{(2)}+1} B_i^{(2)}. \tag{15}$$

Here, the random variables $B_i^{(1)}$ and $B_i^{(2)}$ are independent, exponentially distributed service times with parameter μ. The random variable $Y_1^{(1)}$ is the number of machines in the waiting queue (including the one in repair) at the instant that the tagged machine breaks down. The random variable $Y_1^{(2)} + 1$ equals the random position allocated to the tagged machine in the waiting queue at the instant it is moved from the waiting room to the waiting queue.

This model is not a closed product-form network, so that an exact analysis of the sojourn time is considerably more difficult than the analysis for the models considered above. However, a particularly easy approximation of the first moments can be obtained, if one makes the following two assumptions:

- The two components of S_{GROS} in (15) are uncorrelated.
- The random variables $Y_1^{(1)}$ and $Y_1^{(2)}$ are uniformly distributed on $0, 1, \cdots, Y_1$, where the random variable Y_1 is as defined in Section 3.

Neither assumption is strictly valid; however, for the case considered in which $N\lambda > \mu$ and N large, they appear to be good approximations. Using (15) and the fact that $B_N(\rho) \to 0$ like $\rho^N/N!$ for N large and $N >> \mu/\lambda$, see above (9), it follows that

$$\mathrm{var}(S_{GROS}) \approx \frac{1}{\mu^2}\left[(N - \mu/\lambda)^2/6 + (4N - 2\mu/\lambda)/3\right]. \tag{16}$$

Thus, for large N, the GROS variance is much larger than the variance in the machine repair model with the FCFS service discipline. However, it is considerably smaller than the variance in the machine repair model with the ROS service discipline.

6 A Comparison

We now turn to a comparison of the access delay due to contention resolution and the sojourn time in the variants of the machine repair model. In this comparison, we will confine ourselves to the first two moments of the various distributions: we consider first moments in Section 6.1 and standard deviations in Section 6.2.

The procedures for contention resolution were described in Section 2, and the access delay due to contention resolution is the delay experienced by stations that use contention trees for reservation. More formally, it is defined as the number of tree slots elapsed from the instant a station becomes active until the instant its request is successfully transmitted. As already indicated in Section 2, there are no closed form expressions for the moments of the distribution of the access delay. Hence, these are obtained via simulation. In these simulations, the stations execute the procedure outlined in Section 2. Thus, we use a source model in which each of a finite number, N, of stations generates packets according to a Poisson process with rate λ, independently of the other stations.

The average delays thus obtained are denoted $\widehat{ES_F}$ and $\widehat{ES_B}$, for the 'free' and 'blocked' channel access protocol respectively. Likewise, the estimated standard deviations are denoted by $\widehat{\sigma_F}$ and $\widehat{\sigma_B}$. The 'hat' serves as a reminder that the moments are estimated from a simulation. We use 1000 trees in each simulation.

The moments of the sojourn time of the various machine repair models have been obtained in Sections 3 to 5. In utilizing the results from these sections, we will use $\mu = \log(3)$ for the rate of the service time distribution. The motivation behind this value is in Janssen and de Jong ([14], Eq. 26-27). They show that the average number of nodes to complete a tree with n contenders is well approximated by $n/\log(3)$.

Table 1. Average access delay for reservation with free tree, $\widehat{ES_F}$, with blocked tree, $\widehat{ES_B}$, and expected sojourn time for the machine repair model, ES, for number of stations N, and total traffic intensity Λ

	$N = 100$			$N = 200$			$N = 1000$		
Λ	$\widehat{ES_F}$	$\widehat{ES_B}$	ES	$\widehat{ES_F}$	$\widehat{ES_B}$	ES	$\widehat{ES_F}$	$\widehat{ES_B}$	ES
2.5	43.0	50.1	51.0	86.0	101.3	102.0	429.1	509.4	510.0
5.0	63.0	70.5	71.0	125.9	141.5	142.0	629.9	710.8	710.0
10.0	73.1	80.5	81.0	146.0	161.6	162.0	729.7	811.2	810.0
16.5	77.1	84.5	84.9	154.5	169.5	169.9	824.9	848.5	850.0

6.1 First Moments

The average access delays for the tree models and the expected sojourn time for the machine repair model are given in Table 1. There is only one entry in the table corresponding to the expected sojourn time, as it is the same for all variants of the machine repair model considered. In the table, we have varied the number of stations, N, and the total traffic intensity $\Lambda := N\lambda$. The primary purpose of this table is to compare average access delay with expected sojourn time. Whence, the intensities are chosen so that Λ is well above μ, which is the case most relevant to access in cable networks.

From the figures we can draw various conclusions. Firstly, and most importantly, we observe that the expected sojourn time in the machine repair model provides an excellent approximation to the average access delay for reservation with contention trees. The agreement with the figures obtained via simulations with blocked access is almost perfect; the agreement with the results for free access is less good. The former result is closely related to a result in Denteneer and Pronk [15] on the average number of contenders in a contention tree.

Secondly, we see that free access is a more efficient access protocol than blocked access in that the average access delay with the former is smaller than the average delay with the latter. This result parallels the result for the open model and the Poisson source model, as graphically illustrated in Figure 16 of Mathys and Flajolet [3]. The considered variants of the machine repair model all lead to the same expected sojourn time and are apparently not sufficiently detailed as models to capture the first moment differences between the blocked and the free access protocols.

Finally, we observe that all quantities investigated in Table 1 depend approximately linearly on the number of stations (for the cases with $N >> \mu/\lambda$).

6.2 Standard Deviations

We next turn to a numerical comparison of the standard deviations in the various models. These are given in Table 2, again for different N and Λ.

Table 2. Standard deviations of the access delay for reservation with free tree, $\widehat{\sigma_F}$, with blocked tree, $\widehat{\sigma_B}$, and standard deviations for the basic machine repair model, σ, the ROS machine repair model, σ_{ROS}, and the GROS machine repair model, σ_{GROS} for number of stations N, and total traffic intensity Λ

	$N = 100$					$N = 1000$				
	Tree		Repair			Tree		Repair		
Λ	$\widehat{\sigma_F}$	$\widehat{\sigma_B}$	σ	σ_{ROS}	σ_{GROS}	$\widehat{\sigma_F}$	$\widehat{\sigma_B}$	σ	σ_{ROS}	σ_{GROS}
2.5	46.1	19.5	9.1	50.45	22.8	429.1	185.1	28.8	509.64	210.4
5.0	68.0	26.7	9.1	70.18	30.6	629.9	261.6	28.8	709.39	291.7
10.0	78.4	30.4	9.1	80.13	34.6	729.7	299.2	28.8	809.34	332.4
16.5	83.2	31.5	9.1	84.06	36.2	786.6	310.6	28.8	848.73	348.4

Several conclusions can be drawn from the table. Firstly, we observe that the standard deviation in either tree model changes with traffic intensity and grows approximately linearly with the number of stations. Neither of these properties is captured by the basic machine repair model; there, the standard deviation of the sojourn time is independent of the traffic intensity and grows only with the square root of the number of stations in the model.

Secondly, the standard deviation of the access delay in the *blocked tree* model corresponds closely to the corresponding figure for the GROS machine repair model. The difference between the two standard deviations is approximately 15%. The results for the GROS model capture both the dependence on the traffic intensity and the dependence on the number of machines that is observed in the tree simulations. Similarly, the standard deviation of the access delay in the *free tree* model corresponds closely to the corresponding figure for the ROS machine repair model.

Looking more closely at the results, we see that the standard deviations obtained for the GROS machine repair model are always larger than those obtained in the blocked tree simulations. We consider this as a fundamental limitation of the machine repair model as an approximation. The batch nature of the contention trees implies that it takes some initial time before the first successful request is transmitted. After this initial period, successful transmissions occur fairly uniformly over the length of the trees. Thus the variability of the waiting period is somewhat reduced as compared to the proposed model in which the successful transmissions occur uniformly over the full length of the tree.

Thirdly, the standard deviations with the free access protocol far exceed those with the blocked access protocol. This result has no parallel in the open model. In fact, Figure 17 in Mathys and Flajolet [3] shows that the standard deviation of the delay with free access protocol is *below* the corresponding value with blocked access for most traffic intensities. However, for large traffic intensities just below the stability bound the order reverses and blocked access then results in smaller standard deviations. Of course, our simulations operate at total traffic intensities that exceed the stability bound for the open system.

Summarizing, our numerical experiments show that the expected sojourn time in the repair stage perfectly matches the average access delay for both variants of the tree procedure. The sojourn time variance in the model with ROS service discipline gives a good approximation of the access delay variance when using free trees. Similarly, the sojourn time variance in the model with GROS service discipline gives a good approximation of the access delay variance when using blocked trees. More numerical results are presented in [16].

Acknowledgement. The authors like to thank Marko Boon for doing a major part of the numerical calculations.

References

1. Driel, C-J. van, van Grinsven, P.A.M., Pronk, V., Snijders, W.A.M.: The (r)evolution of access networks for the information super-highway. IEEE Communications Magazine **35** (1997) 2-10
2. Digital Video Broadcasting (DVB); DVB interaction channel for Cable TV distribution systems (CATV), working draft (Version 3), June 28, 2000, based on European Telecommunications Standard 300 800 (March 1998)
3. Mathys, P., Flajolet, Ph.: Q-ary collision resolution algorithms in random-access systems with free or blocked channel access. IEEE Trans. Inf. Theory **31** (1985) 217-243
4. Tsybakov, B.: Survey of USSR contributions to random multiple-access communications. IEEE Trans. Inf. Theory **31** (1985) 143-165
5. Kleinrock, L.: Queueing Systems, Vol. 2. Wiley, New York (1976)
6. Bertsekas, D.P., Gallager, R.G.: Data Networks. Prentice-Hall, Englewood Cliffs, N.J (1992)
7. Capetanakis, J.I.: Tree algorithms for packet broadcast channels. IEEE Trans. Inf. Theory **25** (1979) 505-515
8. Tsybakov, B.S., Mikhailov, V.A.: Random multiple access of packets: Part and try algorithm. Probl. Peredachi Inf. **16** (1980) 65-79
9. Mitra, D.: Waiting time distributions for closed queueing network models of shared-processor systems. In: F.J. Kylstra (ed.), Performance'81, NHPC, Amsterdam (1981) 113-131
10. Kobayashi, H.: Modeling and Analysis. An Introduction to System Performance Evaluation Methodology. Addison-Wesley, Reading (Mass.) (1978)
11. Sevcik, K.C., Mitrani, I.: The distribution of queueing network states at input and output instants, In: M. Arato *et al.* (eds.), Performance'79, NHPC, Amsterdam (1979) 319-335
12. Borst, S.C., Boxma, O.J., Morrison, J.A., Núñez Queija, R.: The equivalence of processor sharing and service in random order. SPOR-Report 2002-01, Eindhoven University of Technology (2002)
13. Cohen, J.W.: On processor sharing and random order of service (Letter to the editor). J. Appl. Probab. **21** (1984) 937
14. Janssen, A.J.E.M., de Jong, M.J.M.: Analysis of contention tree-algorithms. IEEE Trans. Inf. Theory **46** (2000) 2163-2172
15. Denteneer, D., Pronk, V.: On the number of contenders in a contention tree. Proc. ITC Specialist Seminar, Girona (2001) 105-112
16. Boxma, O.J., Denteneer, D., Resing, J.A.C.: Some models for contention resolution in cable networks. EURANDOM Report 2001-037 (2001)

Adaptive Creation of Network Applications in the Jack-in-the-Net Architecture

Tomoko Itao[1], Tetsuya Nakamura[1], Masato Matsuo[1], Tatsuya Suda[2*], and Tomonori Aoyama[3]

[1] NTT Network Innovation Laboratories, Nippon Telegraph and Telephone Corporation (NTT), 3-9-11 Midori-cho, Musashino-shi, Tokyo, 180-8585, Japan
{tomoko, tetsuya, matsuo}@ma.onlab.ntt.co.jp

[2] Information and Computer Science, University of California, Irvine, Irvine, CA 92697-3425, USA
suda@ics.uci.edu

[3] Information and Communication Engineering, The University of Tokyo, 7-3-1 Hongo Bunkyo-ku, Tokyo, 113-8656, Japan
aoyama@mlab.t.u-tokyo.ac.jp

Abstract. The Jack-in-the-Net Architecture (Ja-Net) is a biologically-inspired approach to design adaptive network applications in large-scale networks. In Ja-Net, a network application is dynamically created from a collection of autonomous components called *cyber-entities*. Cyber-entities first establish relationships with other cyber-entities and collectively provide an application through interacting or collaborating with relationship partners. Strength of a relationship is the measure for the usefulness of the partner and adjusted based on the level of satisfaction indicated by a user who received an application. As time progresses, cyber-entities self-organize based on strong relationships and useful applications that users prefer emerge. We implemented Ja-Net platform software and cyber-entities to verify how popular applications (i.e., applications that users prefer) are created in Ja-Net.

1 Introduction

We envision in the future that the Internet spans the entire globe, interconnecting all humans and all man-made devices and objects. When a network scales to this magnitude, it will be virtually impossible to manage a network through a central, coordinating entity. A network must be autonomous and contain built-in mechanisms to support such key features as scalability, adaptability, simplicity,

[*] A part of Tatsuya Suda's research presented in this paper was supported by the National Science Foundation through grants ANI-0083074 and ANI-9903427, by DARPA through Grant MDA972-99-1-0007, by AFOSR through Grant MURI F49620-00-1-0330, and by grants from the University of California MICRO Program, and Nippon Telegraph and Telephone Corporation (NTT).
Tatsuya Suda also holds the title of NTT Research Professor, and his NTT contact information is same as the co-authors' contact information.

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 129–140, 2002.

and survivability. We believe that applying concepts and mechanisms from the biological world provides a unique and promising approach to solving key issues that future networks face.

The Jack-in-the-Net Architecture (Ja-Net)[1][2] is a biologically-inspired approach to design adaptive network applications in future networks. The biological concept that we apply in Ja-Net is *emergent behavior* where desirable structure and characteristics emerge from a group of interacting individual entities. In Ja-Net, a network application is dynamically created from a group of interacting autonomous components called *cyber-entities.* A cyber-entity is software with simple behaviors such as migration, replication, reproduction, relationship establishment and death, and implements a set of actions related to a service that the cyber-entity provides. An application is provided through interactions of its cyber-entities. In providing applications, cyber-entities first establish relationships with other cyber-entities and then choose cyber-entities to interact with based on relationships. Strength of a relationship indicates the usefulness of the partner and dynamically adjusted based on the level of satisfaction indicated by a user who received an application. As time progresses, cyber-entities self-organize based on strong relationships resulting in useful emergent applications that users prefer.

In this paper, we describe design and implementation of mechanisms to create applications adaptively in Ja-Net. The rest of the paper is organized in the following manner. Section 2 describes related work. Section 3 describes the overview of Ja-Net Architecture and design of cyber-entities. Section 4 describes experiments on dynamic creation of applications in Ja-Net. Conclusion and future work are discussed in section 5.

2 Related Work

Currently, some frameworks and architectures exist for dynamically creating applications. One such example is Hive [3], where an application is provided through interaction of distributed agents. In Hive, agents choose agents to interact with by specifying the Java interface object that each agent implements. Thus, interaction in Hive is limited to among the agents that mutually implement the interface object of the partner. Unlike Hive, Ja-Net supports ACL (Agent Communication Language) [4] to maximize flexibility in cyber-entity interactions. Bee-gent [5] is another example of a framework to create applications dynamically. It uses a centralized mediator model; a mediator agent maintains a centralized application scenario (logic) and coordinates agent interactions to reduce complexity in multi-agent collaboration. With this centralized mediator approach, Bee-gent restricts the flexibility and scalability of the agent collaboration. Unlike Bee-gent, in Ja-Net, there is no centralized entity to coordinate cyber-entity services, and thus, it scales in the number of cyber-entities. In addition, Ja-Net goes one step further than these architectures by providing built-in mechanisms to support adaptive creation of network applications that reflect user preferences and usage patterns.

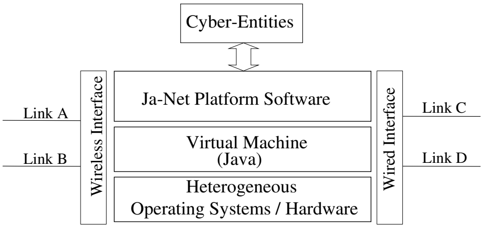


Fig. 1. Ja-Net node structure

Current popular mobile agent systems, including IBM's Aglets [6], General Magic's Odyssey [7], ObjectSpace's Voyager [8] and the University of Stuggart's Mole project [9], adopt the view that a mobile agent is a single unit of computation. They do not employ biological concepts nor take the view that a group of agents may be viewed as a single functioning collective entity.

3 Design of Cyber-Entities

3.1 Overview of the Ja-Net Architecture

Each node in Ja-Net consists of the layers as shown in Figure 1. Ja-Net platform software (referred to as the *platform software* in the rest of the paper) runs using a virtual machine (such as the Java virtual machine) and provides an execution environment and supporting facilities for cyber-entities such as a communication and life-cycle management of cyber-entities. Cyber-entities run atop the platform software. The minimum requirement for a network node to participate in Ja-Net to run the platform software.

A cyber-entity consists of three main parts: *attributes*, *body* and *behaviors. Attributes* carry information regarding the cyber-entity (e.g., cyber-entity ID, service type, keywords, age, etc.). The cyber-entity *body* implements a service provided by a cyber-entity. Cyber-entity *behaviors* implement non-service related actions of a cyber-entity such as migration, replication, relationship establishment and death.

3.2 Cyber-Entity Communication

In order to collectively provide an application by a group of cyber-entities, cyber-entities exchange messages during the execution of cyber-entity services. Upon receiving a message, a cyber-entity interprets the message and invokes an appropriate service action and sends the outcome of the action to another cyber-entity that it interacts with. This, in turn, triggers service invocation of those cyber-entities that receive a message. Cyber-entities may also invoke their services based on an event notification. In Ja-Net, various events may be generated triggered by changes in the network or in the real world (such changes may be captured by sensors).

In Ja-Net, to maximize the flexibility in application creation, we adopt Speech Act based FIPA ACL (Agent Communication Language)[4] with extensions specific to the Ja-Net as a communication language of cyber-entities. In the Ja-Net ACL, we define a small number of communicative acts (such as *request*, *agree*, *refuse*, *inform*, *failure*, *query-if*, *advertise*, *recruit*, and *reward*) to facilitate communication between cyber-entities. *Advertise*, *recruit* and *reward* are not in the FIPA ACL communicative acts and specific to the Ja-Net ACL. They are used during the execution of relationship establishment behavior (please see section 3.4 for relationship establishment behavior). In the Ja-Net ACL, an event notification message is also delivered in ACL using *inform* communicative act. Each ACL message exchanged between cyber-entities contains a communicative act and parameters such as *:receiver*, *:sender*, *:in-reply-to*, *:ontology*, *:sequence-id* and *:content*. *:Receiver* and *:sender*, parameters specify the receiver of the current message and the sender of the current message, respectively. *:In-reply-to* specifies to which message it is replying and is to manage the message exchange flow between cyber-entities. *:Ontology* specifies the vocabulary set (dictionary) used to describe the content of the message. *:Sequence-id* specifies a unique identifier of a message sequence in providing an application . A *sequence-id* is generated by a cyber-entity at the initial point of an application and piggy backed by each ACL message exchanged during the application. *:Content* specifies data or information associated with a communicative act in the message. A *:content* parameter is described with Extensible Markup Language (XML)[10].

3.3 Cyber-Entity Body

A cyber-entity service is implemented as a finite state machine. A cyber-entity may have multiple state models and execute them in parallel. Each state model consists of states and state transition rules. A state implements an atomic service action and message exchanges associated with the action (to allow inputting data to and outputting data from a given action in a given state). A state transition rule associated with a state specifies the next state to transit to. When an action in a given state completes, the current state moves to the next state based on the state transition rule.

In sending the outcome of a service action, a cyber-entity may either respond to a cyber-entity that sent the previous message, or send the message to another cyber-entity (or cyber-entities) by selecting a cyber-entity (or cyber-entities) to interact with using relationship (please see section 3.4 for interaction partner selection mechanism). Upon receiving a message from another cyber-entity, a cyber-entity invokes an appropriate state (action) that can handle the message by examining parameters of the incoming message in the following manner. If the parameter *:in-reply-to* is set in the incoming message, it is in response to a previously transmitted message. In this case, the cyber-entity compares the data type of the *:content* in the incoming message with the input data type required by an action (state) where the previous message was transmitted, and invokes the action if it can take the incoming message as its input. If the parameter *:in-reply-to* of the incoming message is null, the incoming message is the first message

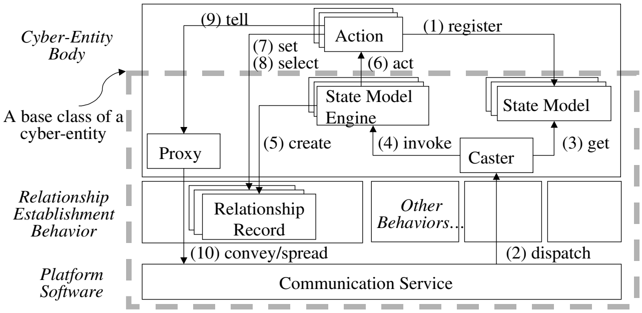


Fig. 2. Function components at a cyber-entity

from the sender cyber-entity. In this case, the receiver cyber-entity examines the current state of a state model that is ready to interact with a new cyber-entity and the initial state of each and every state model that it implements. Among them, a state that can take the incoming message as its input is then invoked.

Figure 2 shows the main function components (classes) of a cyber-entity. In our current design, classes in the cyber-entity *body* except *action* as well as classes in cyber-entity *behaviors* are implemented in a base class of a cyber-entity, and all cyber-entities are derived from the base class. Service actions are implemented by cyber-entity designers and registered with a *state model* (depicted as (1) "register" in Figure 2). *Caster* receives an ACL message from another cyber-entity via the communication service in the platform software (depicted as (2) "dispatch" in Figure 2), examines state models (depicted as (3) "get" in Figure 2) and invokes an appropriate state (action) (depicted as (4) "invoke" and (6) "act" in Figure 2). *State model engine* is a generic class to execute a state model. *Proxy* represents a remote cyber-entity and provides an API to send a message to the remote cyber-entity (depicted as (9) "tell" in Figure 2). The outgoing message is unicast (or multicast)/broadcast by the platform (depicted as (10) "convey/spread" in Figure 2).

3.4 Relationship Management

Relationship Attributes. A relationship may be viewed as (cyber-entity's) information cache regarding other cyber-entities. Table 1 shows example relationship attributes stored in a *relationship record* (depicted as "Relationship Record" in Figure 2) at a cyber-entity. *CE-id* is to uniquely identify a relationship partner cyber-entity. *Action-name* specifies an action of the cyber-entity itself to interact with a relationship partner cyber-entity. *Service-properties* is to store information regarding the service that a relationship partner cyber-entity provides (such as the service type and keywords of a relationship partner cyber-entity). *Access-count* may be incremented when a service message is exchanged with a relationship partner cyber-entity. *Strength* evaluates the usefulness of a

Table 1. Example attributes of a relationship record at a cyber-entity

Attribute	Meaning
CE-id	A globally unique identifier of a relationship partner cyber-entity.
Action-name	An action of the cyber-entity itself that may be used to interact with a relationship partner cyber-entity.
Service-properties	Information regarding the service that a relationship partner cyber-entity provides.
Access-count	The number of interactions with a relationship partner cyber-entity.
Strength	Indication of the usefulness of a relationship partner cyber-entity.

partner cyber-entity and is used to help cyber-entities to select useful interaction partners.

Relationship Establishment. Cyber-entities first establish relationships with other cyber-entities to interact with. For instance, a cyber-entity that has just migrated to a new node may broadcast an *advertise* message specifying information regarding the sender cyber-entity (e.g., service type and/or attributes) to establish relationships with nearby cyber-entities. Upon receiving an *advertise* message, a cyber-entity creates a new *relationship record* (depicted as (5) "create" in Figure 2) and stores the sender cyber-entity's CE-id and information obtained from the incoming *advertise* message in the relationship record. Additional information about a relationship partner cyber-entity obtained through interaction may be stored in the *Service-properties* of its relationship record (depicted as (7) "set" in Figure 2). Alternatively, a cyber-entity may broadcast a *recruit* message specifying conditions on a partner (e.g., service type and/or attributes required for a partner cyber-entity). A cyber-entity that receives a *recruit* message responds with an *inform* message containing its own information if it satisfies conditions specified in the *recruit* message. Through this interaction, the sender cyber-entity and the receiver cyber-entity of the *recruit* message may mutually establish a relationship with each other.

Partner Selection. In selecting an interaction partner cyber-entity (or cyber-entities), a cyber-entity may specify one or more relationship attributes as keys and retrieve its relationship records that match the specified keys (depicted as (8) "select" in Figure 2). If there are multiple relationship records that match the keys, a cyber-entity narrow these relationship records based on relationship strengths so that the cyber-entity interacts more often with cyber-entities with stronger relationships. If there is none or less relationship record that matches the keys, a cyber-entity attempts to discover new cyber-entities by broadcasting an *advertise* message or a *recruit* message to nearby cyber-entities.

Strength Adjustment. In Ja-net, a user indicates in *happiness* the degree of his/her satisfaction with the received application. When a user receives an application, the user creates a *reward* message and sets *happiness* value in the message content. The *reward* is back propagated along the message exchange sequence from a cyber-entity at the end point of the application to a cyber-entity at the initial point of the application. In order to remember a back propagation path, each cyber-entity records the previous and the next cyber-entities in the message sequence along with the corresponding *sequence-id* (which is obtained from ACL *:sequence-id* parameter). Upon receiving a *happiness* value in the *reward* message, each cyber-entity modifies the strength of relationships regarding cyber-entities that it interacted with in providing an application. If a user likes the application, positive *happiness* value is returned and the strength value is increased. If a user dislikes the application, negative *happiness* value is returned and the strength value is decreased. If user is neutral or no *happiness* value is returned, there is no change in the strength value. Therefore, cyber-entities that collectively provide a popular application (i.e., application that a number of users like) will receive a positive *happiness* value more often and strengthen the relationship among themselves, while relationships among cyber-entities that provide a not-so-popular application are weakened.

Group Formation. In order to allow users to explicitly request for an application, cyber-entities collectively providing an application form a group when the relationship strengths among themselves exceed a predetermined threshold value. Once a group is formed, a unique group ID, as well as human-readable application name, is assigned to each group member cyber-entity. Thus, users can request for a group service either by a unique group ID or by a human-readable application (group) name.

4 Experiments on Dynamic Application Creation

In order to verify dynamic creation of applications in Ja-Net, we implemented multiple cyber-entities, as well as platform software, based on the design described in section 3 and performed basic experiments. In our experiments, various realistic scenarios were simulated through running multiple cyber-entities and multiple platform software on computers. Our implementation and experiments are explained below.

4.1 Application Implementation

In applications that we implemented, we consider popular public spots such as the New York City's Times Square, a theater in the nearby Broadway theater district and a cafe on the New York City's Fifth Avenue. A number of people (users) visit these locations, stay there for a while (doing, for instance, window shopping, watching a show, having some coffee at a cafe), and leave. Assume that these users carry a mobile phone or a PDA that is capable of running cyber-entities and communicating with other mobile phones and PDAs in an ad-hoc manner. Assume also that shops in these area implement cyber-entities related to their service and run these cyber-entities on a computer in the shop. In addition, some users may implement their own cyber-entities or have down loaded and carry cyber-entities in their mobile phones and PDAs from where they visited earlier in the day. Various cyber-entities join/leave to/from each location

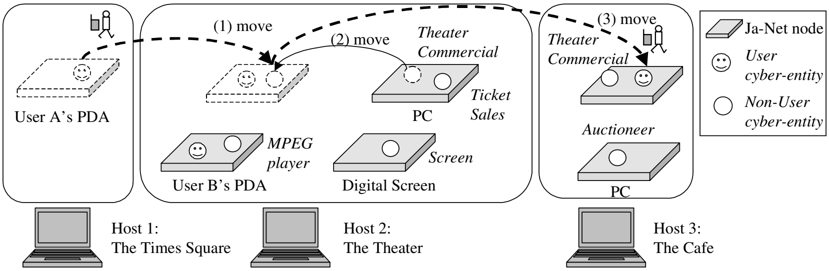


Fig. 3. Overview of the Ja-Net experiment system

according to the movement of users, which triggers actions and interactions of other cyber-entities.

Figure 3 shows the overview of the Ja-Net experiment system. Each host represents different public spot, such as the Times Square, a theater and a cafe, respectively, and each Ja-Net node represents a PC, a device or user's PDA that is supposed to be present at a location that its host computer represents. User's PDA runs a cyber-entity (user cyber-entity) representing the user. Each node runs one or more cyber-entities as described below. User A's PDA at the Times Square runs a user cyber-entity representing user A. A PC at the theater runs a *TheaterCommercial* cyber-entity that stores information of a theater show (assume that this information contains a URL of a commercial video clip of the theater show and a button to request for a ticket purchase as well as other show information) and a *TicketSales* cyber-entity that issues a theater show ticket and generates a certification of ticket purchase. A digital screen in the theater runs a *Screen* cyber-entity that displays image or video on the digital screen. A PDA of another user B at the theater runs a user cyber-entity representing user B and a *MPEGplayer* cyber-entity (assume that it is down loaded by user B earlier in the day). A PC in the cafe runs an *Auctioneer* cyber-entity that purchases commercial products from other cyber-entities and sells them at auction.

In order to capture users' behaviors in our experiments, we defined two types of events. `NODE_ARRIVAL` is an event generated by platform software when a Ja-Net node arrives at a new location. `USER_BROWSING` is an event that is generated by a user cyber-entity when a human user shows interests in the information displayed on his/her PDA. (For instance, this event is generated when a user scrolls the window up and down on his/her PDA). These events are broadcast to cyber-entities in the same location (i.e., in the same host).

Example Application Sequence. Figure 4 shows an message sequence of an application we implemented (referred to as *ticket sales* sequence). In this sequence, a *TheaterCommercial* cyber-entity displays information of a theater show (depicted as (1) "inform" in Figure 4) on user's PDA. Suppose that the user is interested in the show and sends a request for ticket purchase to the *TheaterCommercial* cyber-entity (depicted as (2) "request" in Figure 4). Since the *TheaterCommercial* cyber-entity only stores information of the show and does

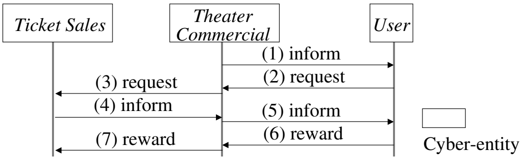


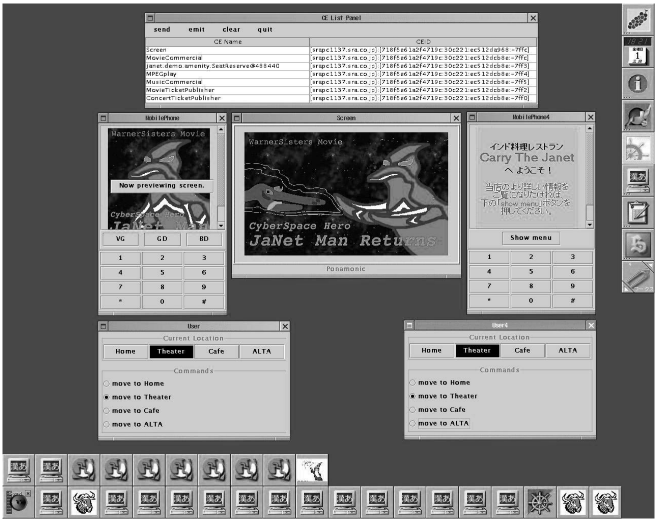
Fig. 4. An example of an application sequence (*ticket sales*)

not implement a ticket sales service, it forwards the request to a *TicketSales* cyber-entity that it has relationship with (depicted as (3) "request" in Figure 4). Upon receiving a forwarded request for a ticket purchase, the *TicketSales* cyber-entity issues a certificate for ticket purchase and sends the certification to the *User* cyber-entity via the *TheaterCommercial* cyber-entity (depicted as (4) "inform" and (5) "inform" in Figure 4 respectively). When a human user obtains a ticket (i.e., a certificate of a ticket purchase) from the *User* cyber-entity, he/she expresses the level of satisfaction as the *happiness* value. *User* cyber-entity then creates a *reward* message and sends it to the *TheaterCommercial* cyber-entity, which, in turn, forwards the *reward* message to the *TicketSales* cyber-entity (depicted as (6) "reward" and (7) "reward" in Figure 4). The relationship strength between the *TheaterCommercial* cyber-entity and the *TicketSales* cyber-entity is adjusted based on the *happiness* value.

4.2 Experimental Results

In our experiments, we only implemented the *body* (i.e., services) and *relationship establishment behavior* of cyber-entities. Thus, cyber-entities were manually moved to simulate their migration behavior when necessary in our experiments. When an experiment starts, cyber-entities initially do not have relationship with any other cyber-entities. Each cyber-entity dynamically establishes relationships with cyber-entities in the same location (i.e., in the same host) by broadcasting an *advertise* message or a *recruit* message. Once relationships are established, cyber-entities start interacting with relationship partners and collectively provide applications. We performed several experiments to examine dynamic application creation in Ja-Net. Our experiments are described below.

Experiment 1. In this experiment, we manually moved a node representing user A's PDA and a user cyber-entity representing user A (on user A's PDA) from the Times Square to the theater (depicted as (1) "move" in Figure3) to simulate user A's movement and observed that an application emerged through interactions of cyber-entities (detailed explanation is described below). Upon arriving at the theater, user A's PDA generated `NODE_ARRIVAL` event and broadcast the event to all cyber-entities in the theater. Upon receiving the event, a user cyber-entity on user A's PDA, one of the cyber-entities in the theater, broadcast an *advertise* message to cyber-entities in the theater. Then, upon receiving the *advertise* message, the *TheaterCommercial* cyber-entity (on PC) established a relationship with a user cyber-entity (on user A's PDA) and sent theater show



CE List Panel

CE Name	CEID
Screen	[srapc1137.sra.co.jp]:[718f6e61a2f4719c:30c221:ec512da968:-7ffc]
MovieCommercial	[srapc1137.sra.co.jp]:[718f6e61a2f4719c:30c221:ec512dcb8e:-7ffc]
janet.demo.amenity.SeatReserve@488440	[srapc1137.sra.co.jp]:[718f6e61a2f4719c:30c221:ec512dcb8e:-7ff3]
MPEGplay	[srapc1137.sra.co.jp]:[718f6e61a2f4719c:30c221:ec512dcb8e:-7ff4]
MusicCommercial	[srapc1137.sra.co.jp]:[718f6e61a2f4719c:30c221:ec512dcb8e:-7ff5]
MovieTicketPublisher	[srapc1137.sra.co.jp]:[718f6e61a2f4719c:30c221:ec512dcb8e:-7ff2]
ConcertTicketPublisher	[srapc1137.sra.co.jp]:[718f6e61a2f4719c:30c221:ec512dcb8e:-7ff0]

インド料理レストラン
Carry The Janet
へ ようこそ！
当店のより詳しい情報を
ご覧になりたければ、
下の「show menu」ボタンを
押してください。

Fig. 5. A screen snap shot of application windows

information that it stores to the user cyber-entity (on user A's PDA), which in turn displayed the theater show information on user A's PDA. At this moment, user A scrolled a window on his/her PDA. (In our experiments, we, human operators conducting the experiment, scrolled a window up and down on user A's PDA). This generated a `USER_BROWSING` event. The event was broadcast to cyber-entities in the theater. In this experiment, we assumed that the *TheaterCommercial* cyber-entity had a relationship with a *MPEGplayer* cyber-entity on a PDA of another user B. Thus, the *TheaterCommercial* cyber-entity, upon receiving the `USER_BROWSING` event, sent theater show information that it stores to the *MPEGpalyer* cyber-entity. The *MPEGplayer* cyber-entity invoked its service and accessed a commercial video clip of a theater show using a URL included in the theater show information. In this experiment, we also assumed that the *MPEGplayer* cyber-entity had a relationship with a *Screen* cyber-entity on the digital screen. Thus, the *MPEGplayer* cyber-entity sent the outcome of its action to the *Screen* cyber-entity. Consequently, the *Screen* cyber-entity displayed the commercial video clip of a show on the digital screen in the theater.

Figure 5 shows application windows displayed by each node. A window of user A's PDA (on the left) displays theater show information and a window of the digital screen (in the center) displays a commercial video clip of a show.

Experiment 2. In this experiment, while theater show information is displayed on user A's PDA, user A clicked a ticket purchase button. (In our experiments, we, human operators conducting the experiment, clicked the button on user A's PDA). A user cyber-entity on user A's PDA generated a *request* message for a ticket purchase and sent it to the *TheaterCommercial* cyber-entity (on PC). Then, we observed interaction between the *TheaterCommercial* cyber-entity and the *TicketSales* cyber-entity (on PC) shown in Figure 4. User A then received a certificate of ticket purchase. At this point, we have demonstrated that an application was created upon receiving a *request* message from a user.

Next, in order to show that different applications emerge in different environments (i.e., environments where different sets of cyber-entities exist), we simulated the *TheaterCommercial* cyber-entity migrated to user A's PDA (i.e., *TheaterCommercial* cyber-entity was manually moved to user A's PDA, which is depicted as (2) "move" in Figure 3), and also simulated user A's movement from the theater to the cafe (depicted as (3) "move" in Figure 3). Upon arriving at the cafe, user A's PDA generated a `NODE_ARRIVAL` event and broadcast the event to cyber-entities in the cafe (including the *TheaterCommercial* cyber-entity on user A's PDA). Upon receiving the event, the *TheaterCommercial* cyber-entity broadcast an *advertise* message to cyber-entities in the cafe. Upon receiving the *advertise* message, the *Auctioneer* cyber-entity (on PC in the cafe) established a relationship with the *TheaterCommercial* cyber-entity and invoked its service action to purchase a commercial product (i.e., a show ticket in this case). Then, a *request* for a ticket purchase is sent from the *Auctioneer* cyber-entity to the *TheaterCommercial* cyber-entity, which in turn forwarded the *request* message to the *TicketSales* cyber-entity (on PC in the theater) following the same sequence shown in Figure 4 except the *Auctioneer* cyber-entity played the role of "User" in this case. The *Auctioneer* cyber-entity received a certificate of ticket purchase and it provided auction service to users by selling the ticket. This experiment verified that the same cyber-entity may provide different applications by interacting with different cyber-entities.

Experiment 3. In order to examine the group formation mechanism proposed in this paper, we artificially created a large number of requests on user A (in the cafe) to purchase a show ticket and sent them to the *TheaterCommercial* cyber-entity (on user A's PDA in the cafe). We assumed in this experiment that user A is satisfied with a ticket purchased from the *TheaterCommercial* cyber-entity, and thus, user A always returned a positive *happiness* value. As time progresses, we observed that the relationship strength from the *TheaterCommercial* cyber-entity to the *TicketSales* cyber-entity (on PC in the theater) as well as the relationship strength from the *TicketSales* cyber-entity to the *TheaterCommercial* cyber-entity gradually increased. When both relationship strengths exceeded a predetermined threshold value, a group of the *TheaterCommercial* cyber-entity and the *TicketSales* cyber-entity was formed. Once a group is formed, the *TheaterCommercial* cyber-entity, an initial point of the application, sent an *advertise* message containing the group ID, and user A was able to invoke the group service by sending a *request* message containing the group ID to the *TheaterCommercial* cyber-entity.

Through experiments 1–3, we verified that through the mechanisms we proposed in this paper, Ja-Net dynamically creates applications that reflect user preferences and usage patterns. Several applications emerged in our experiments, and only popular applications (i.e., applications that users prefer) formed a group.

5 Conclusion and Future Work

The Jack-in-the-Net (Ja-Net) Architecture is a biologically-inspired approach to design and implement adaptive network applications. Ja-Net is inspired by and based on the Bio-Networking Architecture project in University of California, Irvine [11][12]. This paper described design of cyber-entities and key mechanisms used in Ja-Net for cyber-entity interaction and relationship management. This paper also examined and verified these key mechanisms through experiments.

As for future work, we plan to support interaction protocols between cyber-entities to allow more complex collaboration. We also plan to investigate various algorithms for relationship strength adjustment and partner selection in addition to these described in this paper. Various algorithms will be empirically evaluated for their efficiency in creation and provision of adaptive applications. Experimental study through implementation and deployment of a large scale applications will also be conducted.

References

1. T. Suda, T. Itao, T. Nakamura and M. Matsuo, "A Network for Service Evolution and Emergence," Journal of IEICEJ, Invited Paper, Vol.J84-B, No.3, 2001.
2. T. Itao, T. Nakamura and M. Matsuo, T. Suda, and T. Aoyama, "Service Emergence based on Relationship among Self-Organizing Entities," Proc. of the IEEE SAINT2002 (Best Paper), Jan., 2002.
3. N. Minar, M. Gray, O. Roup, R. Krikorian, and P. Maes, "Hive: Distributed Agents for Networking Things," Proc. of the ASA/MA '99, Aug., 1999.
4. Foundation for Intelligent Physical Agents, "FIPA Communicative Act Library Specification, 2000," available at http://www.fipa.org/
5. T. Kawamura, Y. Tahara, T. Hasegawa, A. Ohsuga and S. Honiden, "Bee-gent: Bonding and Encapsulation Enhancement Agent Framework for Development of Distributed Systems," Journal of the IEICEJ, D-I, Vol. J82-D-I, No.9, 1999.
6. D. B. Lange and M. Oshima, "Programming & Deploying Mobile Agents with Java Aglets," Addison-Wesley, 1998.
7. Odyssey Home Page. http://www.genmagic.com/technology/odyssey.html
8. Voyager Home Page. http://www.objectspace.com/products/voyager/
9. Mole Project Home Page, http://inf.informatik.uni-stuttgart.de/ipvr/vs/projekte/mole.html
10. XML web site, http://www.xml.org
11. The BNA Project Home Page. http://netresearch.ics.uci.edu/bionet
12. Michael Wang and Tetsuya Suda, "The Bio-Networking Architecture: A Biologically Inspired Approach to the Design of Scalable, Adaptive, and Survivable/Available Network Applications," Proc. of the IEEE SAINT2001, Jan., 2001.

Anchored Path Discovery in Terminode Routing

Ljubica Blažević, Silvia Giordano, and Jean-Yves Le Boudec

Laboratory for computer Communications and Applications (LCA)
Swiss Federal Institute of Technology, Lausanne (EPFL), Switzerland
{ljubica.blazevic, silvia.giordano,jean-yves.leboudec}@epfl.ch

Abstract. Terminode routing, defined for potentially very large mobile ad hoc networks, forwards packets along anchored paths. An anchored path is a list of fixed geographic points, called anchors. Given that geographic points do not move, the advantage to traditional routing paths is that an anchored path is always "valid". In order to forward packets along anchored paths, the source needs to acquire them by means of path discovery methods. We present two of such methods: Friend Assisted Path Discovery assumes a common protocol in all nodes and a high collaboration among nodes for providing paths. It is a social oriented path discovery scheme. Geographic Maps-based Path Discovery needs to have or to build a summarized view of the network topology, but does not require explicit collaboration of nodes for acquiring path. The two schemes are complementary and can coexist.

1 Introduction

Routing in mobile ad hoc networks (Manets [7]) is already a difficult task when the network size is considerably small, as studied in most of the Manets' protocols. When the network size increases, the routing task becomes too hard to be addressed with traditional approaches. We consider a large mobile ad hoc network, referred to as *terminode* network. Each node, called terminode here, has a permanent End-system Unique Identifier (EUI), and a temporary, location-dependent address (LDA).

Terminode routing [2], which was proposed for coping with this scenario, is a combination of two routing protocols: *Terminode Local Routing (TLR)* and *Terminode Remote Routing (TRR)*. TLR is a mechanism that allows for destinations to be reached in the vicinity of a terminode and does not use location information for taking packet forwarding decisions. It uses local routing tables that every terminode proactively maintains for its close terminodes. In contrast, TRR is used to send data to remote destinations and uses geographic information; it is the key element for achieving scalability and reduced dependence on intermediate systems. TRR default method is *Geodesic Packet Forwarding (GPF)*. GPF is basically a greedy method that forwards the packet closer to the destination location until the destination is reached. GPF does not perform well if the source and the destination are not well connected along the shortest geodesic path. If the source estimates that GPF cannot successfully reach the destination, it uses *anchored paths*. In contrast with traditional routing algorithms, an anchored path does not consist of a list of nodes to be visited for reaching the destination. An anchored path is a list of fixed geographic points, called anchors. In traditional paths made of lists of nodes, if

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 141–153, 2002.

nodes move far from where they were at the time when the path was computed, the path cannot be used to reach the destination. Given that geographic points do not move, the advantage of anchored paths is that an anchored path is always "valid". In order to forward packets along an anchored path, TRR uses the method called *Anchored Geodesic Packet Forwarding (AGPF)*, described in [2]. AGPF is a loose source routing method designed to be robust for mobile networks. A source terminode adds to the packet a route vector made of a list of anchors, which is used as loose source routing information. Between anchors, geodesic packet forwarding is employed. When a relaying terminode receives a packet with a route vector, it checks whether it is close to the first anchor in the list. If so, it removes the first anchor and sends the packet towards the next anchor or the final destination using geodesic packet forwarding. If the anchors are correctly set, then the packet will arrive at the destination with a high probability. Simulation results show that the introduction of the anchored paths is beneficial or the packet delivery rate [2].

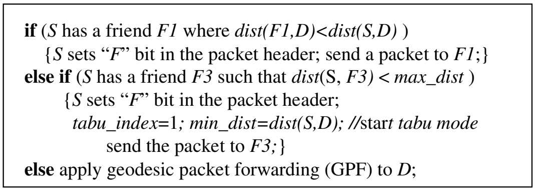
In order to forward packets along anchored paths, the source needs to acquire them by means of path discovery methods. We presented in [2,1] the basic concepts of two such methods: *Friend Assisted Path Discovery (FAPD)*, and *Geographic Maps-based Path Discovery (GMPD)*. FAPD enables the source to learn the anchored path(s) to the destination using, so-called, $friends$, terminodes where the source already knows how to route packets. We describe how nodes select their lists of friends and how these lists are maintained. GMPD assumes that all nodes in the network have a complete or partial knowledge of the network topology. We assume that nodes are always collaborative, that they do not behave maliciously and that they perform protocol actions, whenever requested, in the appropriate way. In this paper we describe FAPD and GMPD.

2 Friend Assisted Path Discovery

FAPD is a default protocol for obtaining anchored paths. It is based on the concept of small-world graphs (SWG) [9]. SWG are very large graphs that tend to be sparse, clustered, and have a small diameter. The small-world phenomenon was inaugurated as an area of experimental study in social science through the work of Stanley Milgram in the 60's. These experiments have shown that the acquaintanceship graph connecting the entire human population has a diameter of six or less; this phenomenon allows people to speak of the "six-degrees of separation". We view a terminode network as a large graph, with edges representing the "friend relationship". B is a $friend$ of A if (1) A thinks that it has a good path to B and (2) A decides to keep B in its list of friends. A may have a good path to B because A can reach B by applying TLR, or by geodesic packet forwarding, or because A managed to maintain one or several anchored paths to B that work well. The value of a path is given in terms of congestion feedback information such as packet loss and delay. Path evaluation is out of the goals of this paper. By means of the TLR protocol, every terminode has knowledge of a number of close terminodes; this makes a graph highly clustered. In addition, every terminode has a number of remote friends to which it maintains a good path(s). We conjecture that this graph has the properties of a SWG. That is, roughly speaking, any two vertices are likely to be connected through a short sequence of intermediate vertices. This means that any two terminodes are likely to be connected with a small number of intermediate friends. With FADP, each terminode keeps the list of its friends with the following information: location of friend, path(s) to friend and potentially some information about the quality of path(s). FAPD is composed by two elements: *Friends Assisted Path Discovery Protocol (FAPDP)* and *Friends Management (FM).*

2.1 Friend Assisted Path Discovery Protocol (FAPDP)

FAPDP is a distributed method for finding an anchored path between two terminodes in a terminode network. When a source S wants to discover a path to destination D, it requests assistance from some friend. If this friend is in condition to collaborate, it tries to provide S with some path to D (it can have it already or try to find it, perhaps with the collaboration of its own friends). Figures 1 and 2 present FAPDP in pseudocode at the source and at an intermediate friend.



```
if (S has a friend F1 where dist(F1,D)<dist(S,D) )
   {S sets "F" bit in the packet header; send a packet to F1;}
else if (S has a friend F3 such that dist(S, F3) < max_dist )
     {S sets "F" bit in the packet header;
        tabu_index=1; min_dist=dist(S,D); //start tabu mode
              send the packet to F3;}
else apply geodesic packet forwarding (GPF) to D;
```

Fig. 1. Friend Assisted Path Discovery Protocol at the source

When source S, which has some data to send to D, has some friends that are closer to D than S itself, it selects friend $F1$ whose location is closest to D, and starts FAPDP with $F1$. S sends the data packet to $F1$ according to the existing path that S maintains to $F1$ because $F1$ is a friend of S. S sets, within the data packet header, the "F" bit[1]. This denotes that the corresponding packet is a *path discovery packet (PDP).*

The *fapd_anchored_path* field inside the path discovery packet progressively contains anchor points from S to D.

If S has an anchored path to $F1$, S simply puts anchors of this path in *fapd_anchored_path* field (S sends data to $F1$ with AGPF). Otherwise, S leaves this field empty (in this case S sends to $F1$ with geodesic packet forwarding). Upon reception of the path discovery packet, $F1$ puts its geographic location (LDA_{F1}) inside *fapd_anchored_path* field as one anchor. If $F1$ has an anchored path to D, $F1$ appends this path into *fapd_anchored_path* field and sends the packet to D by AGPF. If $F1$ does not have a path to D, it recursively uses FAPDP: it checks if it has a friend $F2$ closer to D, and then it performs the same steps as S. This is repeated until the packet is received by some intermediate node that finds D can be reached by means of TLR and it forwards the packet to D by TLR.

[1] the "F" bit is not reset before reaching D

```
F1 is intended receiver of  a path discovery packet ("F"bit = 1 ): S  needs a path to D
if (F1 == D) {send path reply with fapd_anchored_path  to S;}
else if (F1 has a path to D)
  append this path in fapd_anchored_path and send the packet to D;
else if  (tabu_index  > 0 ) //packet in tabu mode
   {
    if ( F1 has a friend F2 where dist(F2, D) < min_dist)
      {tabu_index=0;  send the packet to F2}
    else if (tabu_index < 2 and F1 has a friend F3 such that dist(F1, F3) < max_dist )
          {  tabu_index++;  send a packet to F3}
    else // tabu_index reached the maximum value
         {send a packet to D by geodesic packet forwarding}
   }
else //packet not in tabu mode
 {
  if (F1 has a friend F2 where dist(F2,D)<dist(F1,D) )
   send a packet to F2;
 else if (F1 has a friend F3 such that dist(F1, F3) < max_dist )
      {tabu_index=1; min_dist=dist(F1,D); send a packet to F3}// start tabu mode
 else apply geodesic packet forwarding (GPF) to D;
}
```

Fig. 2. FAPDP at the intermediate friend and at the destination

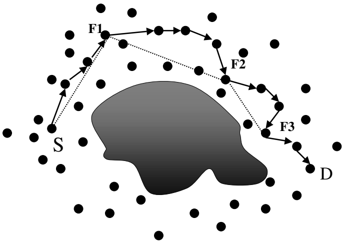


Fig. 3. Figure presents how FAPDP works when source S, has a friend $F1$ that is closer to D than S. S sends data packet to $F1$ and sets the "F" bit in the packet header in order to denote that this is a "path discovery packet". Upon reception of the path discovery packet PDP, $F1$ puts LDA_{F1} inside the $fapd\_anchored\_path$ field of PDP as one anchor. In this example $F1$ does not have path to D, but has a friend $F2$ whose distance to D is smaller than the distance from $F1$ to D. $F1$ sends a PDP to $F2$. In a similar way, $F2$ sends the PDP to its friend $F3$. Once $F3$ receives the PDP, it finds out that D is TLR-reachable and $F3$ forwards the PDP to D by TLR. When D receives the PDP with set "F" bit, it should send back to S a "path reply" control packet with the acquired anchored path from S to D. Assuming that the path from S to $F1$, from $F1$ to $F2$ and from $F2$ to $F3$ does not contain any anchors, the anchored path from S to D is thus a list of anchors $(LDA_{F1}, LDA_{F2}, LDA_{F3})$.

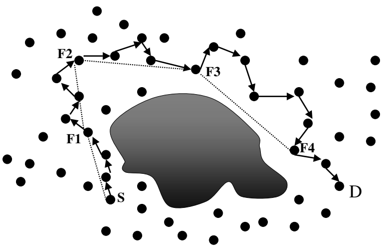


Fig. 4. Figure presents how FAPDP works when source S does not have a friend that is closer to D than itself. S contacts its friend $F1$ that is farther from D in geometrical distance than S is, but such that $dist(S, F1) < max\_dist$. As in the previous example, S sends data packet to $F1$ with "F" bit set. In addition S sets the $tabu\_index$ field to 1 and thus starts the tabu mode of FAPDP. S puts $dist(S, D)$ within min_dist field. Upon reception of the path discovery packet PDP, $F1$ finds out that it does not have a friend whose distance to D is smaller than min_dist. $F1$ forwards the PDP to its friend $F2$ (that is in the opposite direction from D) where $dist(F1, F2) < max\_dist$, and sets $tabu\_index$ to 2. Upon reception of the PDP, $F2$ checks that $tabu\_index$ is equal to its maximum value, and $F2$ cannot forward the PDP to its friend that does not reduce the distance min_dist. In our example, $F2$ has a friend $F3$ whose distance to D is smaller than min_dist and forwards the PDP to it. At $F3$, $tabu\_index$ is reset to 0. This means that FAPDP is not longer in tabu mode. From $F3$ the PDP is forwarded to its friend $F4$ and from there to D by using the TLR protocol. The anchored path from S to D is thus a list of anchors $(LDA_{F1}, LDA_{F2}, LDA_{F3}, LDA_{F4})$

However, there are situations where the source or an intermediate friend does not have a friend closer to the destination. For example, in topologies with obstacles, at some point, going in the direction opposite from the destination may be the only way to reach the destination. Therefore, FAPDP permits to T (the source or an intermediate friend) to send a path discovery packet to a friend even though the packet is not getting closer to the destination. However, such a friend must not be distant from T more than distance max_dist[2]. At that point, it starts the "tabu" mode of FAPDP. When in tabu mode, the packet can be sent in a direction opposite to D for a limited number of times. This is inspired to the Tabu Search heuristic ([4], [5]). Tabu Search can be defined as a general heuristic in which a local search procedure is applied at each step of the general iterative process. It could be superimposed on other heuristics to prevent those being trapped in a local minimum. We use the tabu mechanism in order to get out of a local minimum that can happen at some node that does not have a friend closer to the destination. With the tabu mechanism, we try the opposite direction from the destination with the aim to finally get out of a local minimum and further approach towards the destination. Tabu mode is denoted at T by setting the $tabu\_index$ field inside the packet to 1 (default value of $tabu\_index$ is 0). Tabu mode mechanism uses a field called min_dist, where the terminode that started the tabu mode puts its distance to the destination. When an intermediate friend $F1$ receives the path discovery packet, which is in tabu mode, it first checks if it has a friend whose distance to D is smaller than min_dist. If this is the case, the packet is sent to such a friend, and $tabu\_index$ is reset to 0. Otherwise, $F1$ may

[2] we use max_dist equal to five times the transmission range of a terminode

forward the packet to its friend $F2$ whose distance to D is more than min_dist and $F2$ increments $tabu\_index$. In FAPDP, the number of times that the packet is forwarded to a friend that is further from D than min_dist is limited to two (i.e., the value of $tabu\_index$ must not be larger than two). Tabu mode mechanism stops either because a friend that is a distance from D less than min_dist is found, or because $tabu\_index$ is equal to 2. In the second case the packet is forwarded directly to D by geodesic packet forwarding.

Finally, when D receives the packet with the “F" bit equal to one, D must send back to S a *path reply* control packet with the acquired anchored path from S to D. This packet is sent to S by reverting the anchored path and applying AGPF. Once S receives from D a packet with the anchored path, S stores this path in its route cache. If S does not receive an anchored path within some time, or if S wants more paths to D, S starts FAPDP with some other friend. The example presented in Figure 3 shows the case where the path from S to D is found by using three intermediate friends. The example in Figure 4 illustrates the tabu mode of FAPDP.

2.2 Friends Management (FM)

$FriendsManagement(FM)$ is the set of procedures for selecting, monitoring and evaluating friends. For each node, FM maintains a (fixed-size) set of nodes: the *list of friends*. The list of friends contains the nodes that are contacted with FAPDP for discovering paths. Friends Management consists of the following components: *Friends Monitoring*, *Friends Evaluation, Potential Friends Discovery* and *Friends Selection*. FM is critical in the initial phase (bootstrapping). When a node bootstraps, it does not have any information on (possible) friends. Then, the Potential Friends Discovery component is invoked by a node, with the aim of learning, from other nodes, information about some potential friends. Potential friends are also subject to the Friends Selection action. A number of friends are selected from the list of potential friends, taking into account their geographic positions in order to build a friendship graph with the small world graph properties.

Friends Monitoring and Friends Evaluation. Friends are periodically evaluated in order to assure the consistency of the information on current friends and for testing the validity of these friends. We assume that some form of location tracking is active between friends. The *Friends Monitoring* component of FM keeps under control, for a node A, a set of parameters for each friend F_i of A. This consists of:

- Value of the path(s) to the friend F_i: A may evaluate that the path(s) to its friend F_i, that worked well in the past, deteriorated.
- Location of the friend F_i and the average distance to the friend F_i: F_i may have moved considerably from the location where it was at the time when it was included in A's list of friends.
- The number of times friend F_i was contacted to provide a path and the number of paths that are found with the help of the friend F_i: A may contact F_i in FAPDP to learn the path to a given destination, but the path does not come back to A.

Based on these parameters, the *Friends Evaluation* component periodically evaluates whether it is beneficial to keep a node in the list of friends, or if it is better to discard it. Friends with bad evaluation results are discarded from the friends list. At run-time, initial friends disappear- very likely, in order to be substituted by more valid friends. If the number of friends that remain in a list after evaluation is low, new $potential friends$ can be obtained with the Potential Friends Discovery component.

Potential Friends Discovery. Terminode T can have frequent communications with some other terminodes (e.g., for personal, social, business, and economical interest). These terminodes can be directly selected as friends, because it is of T 's interest to maintain constantly a path to them. However, in general, a node could have a small number of terminodes that it contacts frequently. Therefore, it is necessary an automatic mechanism to select new friends. This is the task of the *Potential Friends Discovery (PFD)* and the *Friends Selection (FS)* components. With the PFD component, T receives information on some possible friends from other nodes in the network. This applies both at the bootstrap phase, and periodically, under request from the friend management component. Terminodes periodically send HELLO messages, for the purpose of building of the TLR routing tables [2]. In this process, can learn about the EUIs and the LDAs of the one-hop and the two-hops distant nodes. Given that this information is periodically maintained, a node has information about close nodes at all time, which can be been considered as *close potential friends*. Potential friends that are further than two hops (i.e. T does not maintain information about their EUIs and the LDAs by means of HELLO messages) are called *remote potential friends*. One way for T to learn about remote potential friends is to extract this information from its previous communications. However, to avoid situations where this implicit discovery would not perform properly (e.g. after a long IDLE or OFF period), this component includes a protocol that enables a node to explicitly discover remote potential friends. In this scheme, each node T sends the $get\_friends\_request$ message towards four geographic points ($GP1$, $GP2$, $GP3$ and $GP4$). These points are randomly selected as four points in orthogonal directions at four times the transmission range of T. Once node Y on the way towards a point GP_i finds that GP_i is reachable with TLR, it stops forwarding the message. Then Y sends back the $get\_friends\_reply$ message to T, which contains the list of friends of Y. If this table is empty, Y puts itself in the content field of the message. When node T eventually receives the $get\_friends\_reply$ message from node Y, it combines the received information with the current one of its potential friends. In the case T does not receive a sufficient number of potential friends as reply, it selects four new orthogonal geographic points and repeats the steps described above. After T acquires a list of potential friends, it applies the FS to select a certain number of friends[3] that it includes in the friends list.

Friends Selection: On How to Build a Small World Graph of Friends. The goal of *Friends Selection* component is that terminodes select their friends in a way such that the resulting friendship graph has the properties of a small world graph, as assumed by

[3] We do not define, in this context, how large is the number of friend that a node maintains in its list. This is matter of ongoing work.

FAPD. In the friendship graph vertices correspond to terminodes and there is the edge between nodes i and j, if node i has as a friend node j. The key to generate the small-world phenomenon is the presence of a small fraction of long-range edges, which connect otherwise distant parts of the graph, while most edges remain local, thus contributing to the high clustering property of the graph. Our strategy is to consider geographic positions of nodes when building friendship connections. We distinguish two types of friendship connections:

- short-range friendship connections (local) correspond to connections to one hop distant terminodes (physical neighbours). These local friendship connections aim to make a friendship graph clustered.
- long-range friendship connections (shortcuts): correspond to “logical" connections to terminodes that are more than one hop distant. Each node chooses a small number of them. A shortcut is represented in a friendship graph as one edge.

Our strategy for choosing long-range contacts is inspired by Kleingberg’s paper [6]. In order to determine its shortcuts, a node takes into consideration the distances from the other nodes in the graph. A node chooses with higher probability friends that are closer to it. However, there is always some probability that it will choose some distant friend. Kleingberg considers a two-dimensional square lattice, where each node is joined to its four nearest neighbours. Then, for each vertex one shortcut is added, but not purely at random. For each vertex, all the possible destinations of a shortcut link are assigned a rank based on their lattice distance from the source vertex. The probability of choosing a vertex at distance d is proportional to d^{-r}, where r is an additional parameter of the model. In the case when $r = 0$, shortcuts are chosen with uniform probability. Then with a high probability, there are paths between every pair of nodes and these paths are bounded by a polynomial in $log(n)$, exponentially smaller than the total number of nodes n. However, there is no way for a decentralized algorithm to find these paths. When r is large, then only close nodes have a chance to be connected with a shortcut. The key value for r turns out to be 2 [6]. When $r = 2$, it is shown that the resulting graph is a small world graph and there is a distributed algorithm for finding short paths between any two vertices (paths are exponentially smaller than the total number of nodes). This algorithm is greedy, described as follows in Kleinberg’s paper: in order to find a path from vertex S to vertex D, S lists all edges that come out of it, and chooses the one that connects S to the vertex that is closest to D, as measured by lattice distance; then repeat the same procedure until D is reached.

Inspired by the Kleinberg’s results, we propose the following. The probability that node X_i selects node X_j as its long-distance friend from the list of potential friends $(X_k, k \in 1..n, k \not= i)$ is given with the formula:

$$p(X_i, X_j) = \frac{1/dist^2(X_i, X_j)}{\sum_{k=1,k\neq i}^{n} 1/dist^2(X_i, X_k)} \quad (1)$$

In this formula the probability for node X_i of choosing a friend X_j is proportional to $d(X_i, X_j)^{-2}$. Long-range friend connections are thus selected at random and are not necessarily bi-directional. X_i may have X_j as a friend, but X_j may not have X_i as a friend.

We used simulations to verify our strategy in selecting long-range friendship connections. Simulations were performed with the following assumptions:

- Nodes in the network are distributed as a two-dimensional Poisson point process with a given density.
- All nodes have the same the transmission range R.
- All nodes have a knowledge of identities and locations of all other nodes.

Initially, a friendship graph contains only short-range connections. There is a short-range connection between X_i and X_j, if dist(X_i,X_j)$\leq$ R. Then, every node selects a number of shortcuts from its list of potential friends. We performed simulations where this number is equal to one or two. In our simulations, a node has in a list of potential friends the whole set of nodes except nodes with whom short-range connections are already established.

The algorithm that node X_i uses to select its friends consists of three steps:

- Step 1: If node X_i keeps n nodes in its list of potential friends, interval $[0,1]$ is divided into n intervals. The length of j^{th} interval is equal to $p(X_i, X_j)$, given by Equation (1).
- Step 2: For each friend to be selected, a random trial is performed: a uniform random deviate r in interval $[0,1]$ is generated. If r falls in the j^{th} interval, X_j becomes a friend of X_i.
- Step 3: The same procedure is repeated for each friend that X_i selects.

The friendship graph is built when all nodes select their friends. Then, we find the *characteristic path length (CPL)* of a graph is the median of the means of the shortest path length connecting each vertex to all other vertices. CPL in a way presents the typical shortest path length between every vertex and every other vertex. CPL is also used by Watts in [9] as a metric to verify whether a graph is a small world graph, with thus a small CPL.

With simulations we want to verify that adding a small number of shortcuts in a friendship graph reduces the CPL of the graph, and that a greedy algorithm for finding paths succeeds in finding short paths. With the greedy algorithm, the source and every intermediate node forward the packet to their short or long-range friend that is closest to the destination. FAPDP is basically a greedy algorithm, and uses the “tabu" mode of operation only when the greedy forwarding is not possible. In our simulation we also calculate CPL, where instead of shortest paths between nodes, we use paths lengths obtained by the greedy algorithm.

Simulation results are given in Figure 5, averaged over ten realizations of random graphs for a given number of nodes. In our simulations, transmission range is (R) is 250 meters. Node density is such that every node has an average of ten neighbours (short-range friendship connections). As we increase the number of nodes, we increase the simulation area, but we keep the same density of nodes. The chosen density ensures that the greedy algorithm succeeds in finding most of the paths. We verified that for less than 5% of pairs of nodes, the greedy algorithm is not able to find a path.

Our simulations have shown the following. First, CPL of the friendship graph exhibits logarithmic length scaling with respect to number of nodes in the graph (see Figure 5).

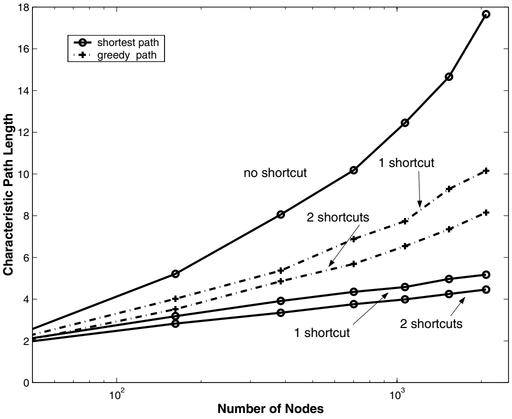


Fig. 5. Figure presents the characteristic path length (CPL) of friendship graph for different number of nodes. Adding a small number of long-range shortcut edges in the graph reduces CPL.

This is a property of a small world graph. A small number of long-range connections (e.g., 1 or 2) are enough to reduce CPL considerably from the case when shortcuts are not used. Second, the greedy algorithm for finding paths succeeds in finding paths whose length is close to shortest paths.

3 Geographic Map-Based Path Discovery (GMPD)

We believe that a good model of a large mobile network does not assume that nodes are uniformly distributed in the network. In order to model a terminode network, we identify the areas with a higher node density, which we call $towns$. Two towns are interconnected by all the nodes in between them (we call it a $highway$). If two towns are interconnected with a highway, there is a high probability that there are terminodes to ensure connectivity from one town to another. GMPD assumes that each terminode has such a summarized geographic view of the network: each terminode has a knowledge of the map of towns. This map defines the town areas and reports the existence of highways between towns. As a first attempt, we model a town area as a square centered in a geographic centre. For each town, the map gives the position of its centre and the size of the square area. The map of a network can be presented as a graph with nodes corresponding to towns and edges corresponding to highways, see Figure 6. Macroscopically, the graph of towns does not change frequently.

GMPD with a given map of towns works as follows:

- Source S determines from its own location (LDA_S) the town area (ST) in which S is situated (or, the nearest town to LDA_S if it is not in the town area). In addition, since S knows the position of destination D (LDA_D), it can determine the town area DT where D is situated (or, the nearest town to LDA_D if it is not in the town area).

- Then, S accesses the network map in order to find the anchored path from S to D. We call this operation a *map lookup*. An anchored path is the list of the geographical points: the points correspond to centers of the towns that the packet has to visit from ST in order to reach DT. One possible realization of the map lookup operation is to find a list of towns that are on the shortest path from ST to DT in the graph of towns; the length of a path can be given either as the number of towns between ST and DT, or the length of the topological (Euclidean) shortest path connecting ST and DT in a graph of towns.

3.1 GMPD with No Initial Summarized View of the Network

Here, we still assume the model of a network based on towns and highways, however, nodes in the network have a constrained view of a network.

A terminode initially does not have the knowledge of a map of towns. The information that a terminode has is the following: 1) if a terminode is within a town area, it knows about neighboring towns areas and town centers with which its current town is connected by highways; 2) if a terminode is on the highway, it knows towns that this highway connects. We assume that a graph of towns is planar. As above, we assume that there is a mapping between the location of a terminode and the corresponding town. Thus, source S can determine from LDA_S the town area (ST) in which S is situated (or, the nearest town to LDA_S if it is not in the town area). In addition, S determines from the LDA_D the town area DT where D is situated (or, the nearest town to LDA_D if it is not in the town area).

Here we present the description of the path discovery algorithm with these assumptions. The source is in town ST and the destination is in town DT.

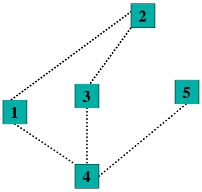


Fig. 6. Figure presents one example of a map of a terminode network. Five town areas (1, 2, 3, 4 and 5) are presented with shaded squares. A highway between two towns is presented with a line between two town areas.

- if ST and DT are the same, then S does not perform path discovery; S sends the packet to D by GPF. If ST and DT are not the same, S begins anchored path discovery. S uses a greedy method: it sends the path discovery packet towards a neighboring town NT whose center is closest to DT. S sends the path discovery packet by using GPF towards the center of NT. As soon as the path discovery packet is received by some terminode N in NT, N adds the center of NT as one anchor in the accumulated anchored path. In addition, N adds NT in the accumulated list of towns (the list of towns is used to record towns that the path discovery packet has visited. This list is later used to simplify the path from ST to DT). If N has a path to DT, it adds this path to the the accumulated anchored path and applies this path to send the packet to D. Otherwise, N repeats the same steps as S.

- if S or an intermediate node (that has to determine the next town to which to send the path discovery packet) does not find a neighboring town closer to DT, the path discovery packet is sent in $perimeter$ mode. In this case, a planar graph traversal method [3] is used, as explained in the example below. This means that next town to send the packet to is determined by using planar graph traversal method applied on a graph of towns.
  As soon as the path discovery packet arrives at some town that is closer to DT than the town where the planar graph traversal method is started, the greedy method resumes in order to find the next town to send the path discovery.
- the explained procedure is repeated until the packet is received by some node in DT. Then the packet is sent directly to D by GPF. When D receives the packet, it analyzes the path and filters possible loops. Then the path is sent back S.
- when S receives the path to DT it adds this path in the list of anchored paths.

We illustrate the path discovery with a localized view of the network with one example. Assume in Figure 6 that source S is in town 1, and destination D is in town 5. S chooses to send the path discovery packet towards the center of town 2 because the center of town 2 is a neighboring town closest to town 5. The first node in town 2 that receives the path discovery packet puts the center of town 2 as one anchor in the accumulated anchored path. Now, because the packet cannot be forwarded closer to town 5, the planar graph traversal algorithm is used. This algorithm is applied on the planar graph of the network map. Following the right hand rule, the first edge of the graph in the direction counterclockwise from the line connecting town 2 and town 5 is the edge that connects town 2 to town 1. Therefore, the packet is sent again to town 1. The planar graph traversal is continued and from town 1 the packet is forwarded to town 4, and then to town 5. As soon as some terminode in town 5 receives the packet it sends it directly to D by using GPF. When D receives the path discovery packet, the accumulated town list is $\{$1,2,1,4,5$\}$. D simplifies the list such that the same town is not visited more than once. The anchored path to be returned to S is the list of towns $\{$1,4,5$\}$ centers.

4 Conclusion

Routing based on anchored paths (AGPF) is a scheme proposed for routing in very large mobile ad hoc networks. Simulation results shows that AGPF is beneficial when there are holes in terminodes distribution and the source cannot reach the destination over the direct geodesic path [2]. This paper presents two schemes for discovering anchored paths, which completes the whole picture. These two schemes, Friend Assisted Path Discovery (FAPD) and Geographic Maps-based Path Discovery (GMPD), are based on two complementary approaches: the social one, e.g. the small world graph approach and the topological one, based on a summarized view of the network. With the most suitable of the two scheme, or with both, a source is able to discover an anchored path to any destination, and thus use the AGPF. We also demonstrated that the resulting friendship graph obtained with FAPD has the wished properties of a small world graph, and illustrated, by examples, how the schemes work. The implementation in the GloMoSim simulator [8] and the further simulation are matter of ongoing work.

References

1. L. Blazevic, S. Giordano, and J.-Y. Le Boudec. Self Organized Routing in Wide Area Mobile Ad-Hoc Network. In *IEEE Symposium on Ad-Hoc Wireless Networks (Globecom 2001)*, November 2001.
2. L. Blazevic, S. Giordano, and J.-Y. Le Boudec. Self Organized Terminode Routing. *Cluster Computing Journal*, April 2002.
3. P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia. Routing with guaranteed delivery in ad hoc wireless newtorks. *3rd Int. Workshop on Descrete Algorithms and methods for mobile computing communications (DIAL M)*, August 1999.
4. F. Glover. Future paths for integer programming and links to artificial intelligence. *Computers and Operational Research*, 13, 1986.
5. P. Hansen. The steepest ascent mildest descent heuristic for combinatorial programming. *Proc. Congr. on Numerical Method in Combinatorial Programming*, Academic Press, 1986.
6. Jon Kleinberg. The small-world phenomenon: an algorithmic perspective. *Technical Report 99-1776, Cornell Computer Science*, 1999.
7. Mobile Ad-hoc Networks (MANET) Working Group. http://www.ietf.org/html.charters/manet-charter.html.
8. M. Takai, L. Bajaj, R. Ahuja, R. Bagrodia, and M. Gerla. GloMoSim:A Scalable Network Simulation Environment. *Technical Report 990027, UCLA, Computer Science Department*, 1999.
9. D. J. Watts. In *Small Worlds, The dynamics of networks between order and randomness*. Princeton University Press, 1999.

Distributed Transmission Scheduling Using Code-Division Channelization*

Lichun Bao and J.J. Garcia-Luna-Aceves[1]

School of Engineering, University of California Santa Cruz, CA 95064, USA
baolc, jj@soe.ucsc.edu

Abstract. We present the Hybrid Activation Multiple Access scheduling protocol (HAMA) for wireless ad hoc networks. Unlike previous channel access scheduling protocols that activate either nodes or links only, HAMA is a node-activation oriented channel access scheduling protocol that also maximizes the chance of link activations. According to HAMA, the only required information for scheduling channel access at each node is the identifiers of neighbors within two hops. Using this neighborhood information, multiple winners for channel access are elected in each contention context, such as a time slot in a frequency band or a spreading code. Except for time slot synchronization and neighbor updates on the two-hop neighborhood changes, HAMA dedicates the bandwidth completely to data communication. The delay and throughput characteristics of HAMA are analyzed, and its performance is compared with pure node activation based scheduling protocols by simulations.

1 Introduction

In ad hoc networks, a channel access scheme usually takes one of two approaches: on-demand or by-schedule. The on-demand approach started with ALOHA and CSMA [9] and continued with several collision avoidance schemes (e.g., MACA [8], FAMA [5] and others). However, as traffic load increases, network throughput drastically degrades because the probability of collisions rises, preventing any station from acquiring the channel. Furthermore, random access MAC protocols cannot provide end-to-end quality assurance due to adverse channel conditions, such as near-far phenomena, and capture effects on the channel.

On the other hand, scheduled access schemes prearrange or negotiate a set of timetables for individual nodes or links, such that the transmissions from these nodes or on these links are collision-free in the code, time, frequency or space divisions of the channel. Collision-free channel access scheduling is typically treated as a node or link coloring problem on graphs representing the network topologies. Searching for an optimal channel access scheduling results in NP-hard problems in graph theory (such as k-colorability on nodes or edges) [3,

* This work was supported in part by Advanced Technology Office of the Defense Advanced Research Projects Agency (DARPA) under grant No. DAAD19-01-C-0026, and by the U.S. Air Force/OSR under grant No. F49620-00-1-0330.

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 154–165, 2002.

4,13]. Polynomial algorithms are known to achieve suboptimal solutions using randomized approaches or heuristics based on such graph attributes as the degree of the nodes.

A unified framework for TDMA/FDMA/CDMA channel assignments, called UxDMA, was described by Ramanathan [12], which combines the general approaches of many other channel access scheduling algorithms in that these algorithms are now represented by UxDMA with different parameters. The parameters to UxDMA indicate the constraints put on the graph entities (nodes or links) such that entities related by the constraints are colored differently. The results of color assignments correspond to channel assignments to these nodes or links in the time, frequency or code domain.

However, since the global topology is required to derive the channel access schedule, topology information needs to be collected and frequent schedule broadcasts have to be carried out in mobile networks, which would consume a significant portion of the scarce wireless bandwidth.

We propose a new channel access scheduling protocol, called HAMA (Hybrid Activation Multiple Access scheduling), that supports both broadcast, multicast and unicast communications in wireless networks using a time and code division multiple access scheme. HAMA establishes channel access schedules directly on local topologies of a node within two hops, which avoids the schedule exchanges and resolutions present in other scheduling algorithms.

Section 2 describes the assumptions we make about multihop wireless networks, and section 3 specifies HAMA. Section 4 presents the neighbor protocol for propagating and acquiring two-hop neighbor information. Section 5 analyzes the delay and throughput attributes of HAMA using queuing theory, and HAMA's performance is studied by simulations and comparisons with NAMA, the pure node-activation based channel access scheduling protocol, and UxDMA, the best-performing scheduling algorithm known to date. Section 6 concludes the paper.

2 Assumptions

The creation and analysis of a packet radio network usually depend on the physical technologies provided and the scenarios where the multihop wireless network is deployed. We assume that:

- Each node in the network is mounted with an omni-directional radio transceiver, and assigned a unique ID number;
- The radio of each node may only work in half-duplex mode, i.e., either transmit or receive data packet at a time, but not both;
- Time is synchronized at each node, and nodes access the channel based on slotted time boundaries. Each time slot lasts long enough to transmit a complete data packet, and is identifiable relative to a consensus starting point.

We do not address the time synchronization issue, but suggest achieving it by attaching timing information in physical layer packet framing and listening to data traffic in the network to align time slots to the latest starting point of a complete packet transmission by one-hop neighbors [10].

The topology of a packet radio network is represented by an undirected graph $G = (V, E)$, where V is the set of network nodes, and E is the set of links between nodes. If link $(u, v) \in E$, then $(v, u) \in E$, and node u and v are within the transmission range of each other, so that they can exchange packets via the common channel, in which case u and v are called *one-hop neighbors* of each other. The set of one-hop neighbors of a node i is denoted as N_i^1.

Since the same codes on a set of the appropriately selected frequency bands can be equivalently considered to be different codes, we only consider channel scheduling based on a time-slotted code division multiple access scheme. Before describing a neighbor protocol in section 4, we assume that each node already knows its two-hop neighbor information, which is the one-hop neighbors of the node itself and the one-hop neighbors of its one-hop neighbors. That is, the collection of neighbor information gathered at node i is:

$$N_i^1 \cup \Big(\bigcup_{j \in N_i^1} N_j^1 \Big) \ .$$

3 HAMA

3.1 Code Assignment

HAMA is a time-slotted code division multiple access scheme using direct sequence spread spectrum (DSSS) transmission techniques. In DSSS, code assignments are categorized into transmitter-oriented code assignment (TOCA), receiver-oriented code assignment (ROCA), or a per-link-oriented code assignment (POCA) schemes [6,7,11]. HAMA adopts transmitter-oriented code assignment (TOCA), because of its broadcast capability.

We assume that a pool of well-chosen quasi-orthogonal pseudo-noise codes, $C_{pn} = \{c^k\}$, are available for each node to choose from. The codes are identified by the superscript $k = 0,\ 1,\ 2,\ \cdots,\ |C_{pn}| - 1$.

The code for each node is denoted by $i.$TxCode $(\in C_{pn})$, and is computed in every time slot. Consequently, a node has varying contention situations for transmission from time slot to time slot. Eq. (1) computes the pseudo-noise code for node i at time slot t.

$$i.\text{TxCode} = c^k,\ k = \texttt{Hash}(i \oplus t) \ \texttt{mod}\ |C_{pn}| \ . \tag{1}$$

where $\texttt{Hash}(x)$ is an integer pseudo-random number generator that generates the message digest of byte-stream input x. The sign '$\oplus$' is designated as the mathematical operation to carry out concatenation on its two operants.

Because we have a limited number of pseudo-noise codes for assignment in HAMA, it is possible that multiple nodes share the same code. We resolve possible collisions using a contention resolution algorithm in HAMA.

3.2 Nodal Modes and Operations

The state of a node is related with a dynamic priority assigned to the node in each time slot. The priority of a node $i \in V$ is computed according to Eq. (2), given the current time slot t:

$$i.\text{prio} = \texttt{Hash}(i \oplus t) \ , \tag{2}$$

where the same function `Hash(x)` was used in Eq. (1).

The mode of a node is determined by the priorities of itself and the surrounding neighbors. We differentiate nodes in terms of their capabilities for transmissions in either broadcast or unicast mode. In general, we allow data packets transmitted from nodes with higher priorities to nodes with lower priorities. Accordingly, the mode of a node is defined into six states under two categories:

- Receiver:
  1. *Sniffer* (Sf): The node does not have the highest priority among its one-hop neighbors.
  2. *Sink* (S): The node has the lowest priority among its one-hop neighbors.
- Transmitter:
  1. *B-Transmitter* (BT): The node has the highest priority among its two-hop neighbors, and can broadcast to its one-hop neighbors without contentions from its two-hop neighbors.
  2. *U-Transmitter* (UT): The node has the highest priority among its one-hop neighbors only. Therefore, the node may transmit to some one-hop neighbors, but cannot do so to other one-hop neighbors.
  3. *S-Transmitter* (ST): The node has the highest priority among the one-hop neighbors of a *Sink* neighbor.
  4. *Yield* (Y): The node could have been either a *B-Transmitter* or a *U-Transmitter*, but chooses to abandon channel access if its transmission may result in hidden-terminal interference at its one-hop neighbors.

If a node i determines that is a transmitter (BT, UT or ST), it prepares data flows for transmissions on its assigned code. The node has to select corresponding one-hop neighbors that can receive its packets. For convenience, we denote the receiver set by $i.\text{out}$, and the packets stored for the eligible receivers by $i.\text{Q}(i.\text{out})$.

If a node i happens to be in reception mode (Sf or S), it chooses a neighbor, denoted by $i.\text{in}$, which has the highest priority among its one-hop neighbors, and listens on the transmission code assigned to $i.\text{in}$.

The transmission code is assigned to a node in transmission mode using Eq. (1), and the reception code of a node i, denoted by $i.\text{RxCode}$, is naturally aligned to transmission code of $i.\text{in}$.

HAMA applies a novel neighbor-aware contention resolution algorithm (NCR) [1] to compute the channel access schedules in each time slot. Provided that each node obtains the accurate knowledge of its neighbors within two hops through neighbor protocol, HAMA decides whether a node i transmits or receives a packet in time slot t on an appropriate code as presented in Fig. 1.

```
HAMA($i$, $t$)
{
    /* Initialized to listen. */
    $i$.mode = Sf;
    $i$.in = -1; /* Null. */
    $i$.out = $\emptyset$; /* Empty set. */

    for ($k \in N_i^1 \cup (\bigcup_{j \in N_i^1} N_j^1)$) {
        $k$.prio = Hash($t \oplus k$);
        $code$ = $k$.prio mod $|C_{pn}|$;
        $k$.TxCode = $c^{code}$;
    }

    for ($\forall j \in N_i^1 \cup \{i\}$) {
        if ($\forall k \in N_j^1$, $j$.prio > $k$.prio)
            /* May unicast. */
            $j$.mode = UT;
        elseif ($\forall k \in N_j^1$, $j$.prio < $k$.prio)
            /* A sink. */
            $j$.mode = S;
    }

    /* More findings about i. */
    if ($i$.mode ≡ UT and
        $\forall k \in \cup_{j \in N_i^1} N_j^1$, $k \;/\!\!\neq i$,
        $i$.prio> $k$.prio)
        /* Can broadcast. */
        $i$.mode = BT;
    elseif ($i$.mode ≡ Sf and
        $\exists j \in N_i^1$, $j$.mode ≡ S and
        $\forall k \in N_j^1$, $k \;/\!\!\neq i$, $i$.prio > $k$.prio) {
        /* Can unicast to a sink */
        $i$.mode = ST;

        /* i has to listen to j. */
        if ($\exists j \in N_i^1$, $j$.mode ≡ UT and
            $\forall k \in N_i^1$, $k \;/\!\!\neq j$, $j$.prio > $k$.prio)
            $i$.mode = Sf;
    }

    /* Determine dest or src. */
    switch ($i$.mode) {
        case BT:
            for ($j \in N_i^1$)
                $i$.out = $i$.out $\cup \{j\}$;

        case UT:
            for ($j \in N_i^1$)
                if ($\forall k \in N_j^1$, $k \;/\!\!\neq i$,
                $i$.prio > $k$.prio)
                    $i$.out = $i$.out $\cup \{j\}$;

        case ST:
            for ($j \in N_i^1$)
                if ($j$.mode ≡ S and
                    $\forall k \in N_j^1$, $k \;/\!\!\neq i$,
                    $i$.prio > $k$.prio)
                    $i$.out = $i$.out $\cup \{j\}$;

        case S, Sf:
            if ($\exists j \in N_i^1$ and
                $\forall k \in N_i^1$, $k \;/\!\!\neq j$,
                $j$.prio> $k$.prio) {
                $i$.in = $j$;
                $i$.RxCode = $j$.TxCode;
            }
    }

    /* Hidden-Terminal Avoidance. */
    if ($i$.mode ∈ { UT, ST } and
        $\exists j \in N_i^1$, $j$.mode /= UTand
        $\exists k \in N_j^1$, $k$.prio > $i$.prio and
        $k$.TxCode ≡ $i$.TxCode)
        $i$.mode = Y;

    /* Ready to communicate. */
    if ($i$.mode ∈ {BT, UT, ST } and
        $i$.Q($i$.out) $/\!\!\neq \emptyset$) {
        /* FIFO */
        $pkt$ = Dequeue($i$.Q($i$.out));
        Transmit $pkt$ on code $i$.TxCode;
    }
    else
        Listen on code $i$.RxCode;
} /* End of HAMA. */
```

Fig. 1. HAMA Specification

4 Neighbor Protocol

In HAMA, topology information within two hops of a node plays a essential role for channel access operations. In mobile networks, network topologies change frequently, which affects the transmission schedules of the mobile nodes. The ability to detect and notify such changes promptly relies on the neighbor protocol so as to reharmonize channel access scheduling.

4.1 Signals

Since HAMA adopts dynamic code assignment for channel access using the identifier and time slot number, it is impossible for a node to detect a new one-hop neighbor that transmits data packets with varying codes. We have to use an additional time section, called *neighborhood section*, for sending out "hello" messages and for mobility management purposes. The neighborhood section lasts for T_{nbr} time slots following every T_{hama} HAMA time slots. Channel access is still based on code division scheme but the transmission code is fixed over a commonly known one selected from C_{pn}.

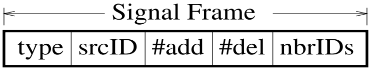


Fig. 2. Signal Frame Format

In addition, a time slot within the neighborhood section is further divided into a number of smaller time segments fit for transmitting short signal packets, where their format is as illustrated in Fig. 2.

In Fig. 2, the signal frame transmitted by node `srcID` is indicated by its `type` field. And field `#add` and `#del` count the numbers of the following nbrIDs for addition and deletion, respectively, of the neighbors from the transmitter's neighbor topology.

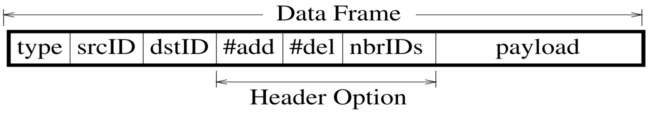


Fig. 3. Data Frame Format

Besides signals, one-hop neighbor updates are also propagated using broadcast packets if a node is activated in BT-mode, so that the update information of a node gets to all its neighbors efficiently. One-hop neighbor updates are piggyback in the option field of a data frame whenever possible and necessary. Fig. 3

illustrates the data packet format, which includes similar neighbor update fields as in Fig. 2, besides regular fields such as destination `dstID` and payload of the packet.

4.2 Mobility Handling

Signals are used by the neighbor protocol for two purposes. One is for a node to say "hello" to its one-hop neighbors periodically in order to maintain connectivity. The other is to send neighbor updates when a neighbor is added, deleted or needs to be refreshed. In case of a new link being established, both ends of the link need to notify their one-hop neighbors of the new link, and exchange their complete one-hop neighbor information. In case of a link breaking down, a neighbor-delete update needs to be sent out. An existing neighbor connection also has to be refreshed periodically to the one-hop neighbors for robustness. If a neighbor-delete update is not delivered to some one-hop neighbors, those neighbors age out the obsolete link after a period of time.

However, because of the randomness of signal packet transmissions, it is possible for a signal sent by a node to collide with signals sent by some of its two-hop neighbors. Due to the lack of acknowledgments in signal transmissions, multiple retransmissions of the update information are needed for a node to ensure the delivery of the message to its one-hop neighbors. Signal intervals also jitter by a small value so that signals transmitted in the neighborhood spread out evenly over the neighborhood section to avoid collisions.

Furthermore, retransmissions of a signal packet can only achieve a certain probability of successful delivery of the message. Even though the probability approaches one as the retransmissions are carried out repetitively, the neighbor protocol has to regulate the rhythm of sending signals, so that the desired probability of the message delivery is achieved with a small minimum number of retransmissions in the shortest time, thus incurring the least amount of interference to other neighbors' signal transmissions.

The number of signal retransmissions and the interval between retransmissions depends on the number of two-hop neighbors. The more neighbors a node has, the longer the interval value is chosen for signal retransmissions. Since the probability of each signal transmission trial can be determined by the interval value, the number of retransmissions can be derived to achieve the desired probability of successful message delivery.

Consequently, the latency of the message delivery using retransmission approach is decided by the product of the interval and the number of retransmissions. If we do not depend on the neighbor updates transmitted in the option field of data frames, enough time slots should be allocated to the neighborhood section in the time division scheme to achieve the desired latency of message delivery, which determines the ratio between T_{hama} and T_{nbr}. Since the time division is fixed during the operations of HAMA, the ratio $T_{hama} : T_{nbr}$ is computed beforehand in the neighbor protocol to handle networks with moderate density. We do not specify the relations from this aspect in this paper.

5 Performance

5.1 Delay Analysis

When data packets arrive at a node according to a Poisson process with rate λ and are served according to the first-come-first-serve (FIFO) strategy, we can analyze the delay properties of HAMA by a steady-state M/G/1 queues with server vacations, where the single server is the node. The server takes a vacation for V of one time slot when there is no data packet in the queue; otherwise, it looks for the next available time slot to transmit the first packet waiting in the queue.

Because a node accesses the channel in a time slot by comparing the random priorities assigned its one-hop neighbors, the attempt of channel access in each time slot is a Bernoulli trial for each node. Depending on the neighborhood topology, each node has a probability q of winning the channel access contention in each time slot. Therefore, the service time of a data packet in the queuing system is a random variable following geometric distribution with parameter q. Denote the service time as X, we have $P\{X = k,\ k \geq 1\} = (1-q)^{k-1}q$.

The mean and second moments of random variable X and V are:

$$\overline{X} = \frac{1}{q}\ , \quad \overline{X^2} = \frac{2-q}{q^2}\ ;$$

$$\overline{V} = \overline{V^2} = 1\ .$$

So that the extended Pollaczek-Kinchin formula for the waiting time in the M/G/1 queuing system with server vacations [2]

$$W = \frac{\lambda \overline{X^2}}{2(1-\lambda\overline{X})} + \frac{\overline{V^2}}{2\overline{V}}$$

readily yields the average waiting period of a data packet in the queue as:

$$W = \frac{\lambda(2-q)}{2q(q-\lambda)} + \frac{1}{2}\ .$$

Adding the average service time to the queuing delay, we get the overall delay in the system:

$$T = W + \overline{X} = \frac{2+q-2\lambda}{2(q-\lambda)}\ . \qquad (3)$$

To keep the queuing system in a steady state without packet overflow problems, it is necessary that $\lambda < q$.

Since HAMA is capable of both node activation and link activation, the delays of broadcast and unicast traffics should be considered separately because the contenders of node activation and link activation are different, so are the respective activation probabilities.

5.2 Throughput Analysis

Network throughput is defined as the number of packets going through the network at the same time including both broadcast and unicast traffics. On account of the collision freedom in HAMA, the shared channel can serve certain load up to the channel capacity allowed without degradations. That is, the throughput over the common channel is the summation of arrival rates at all network nodes as long as the queuing system at each node remains in equilibrium on the arrival and departure events. Therefore, the system throughput S is derived as:

$$S = \sum_{k \in V} \min(\lambda_k,\ q_k) \ , \tag{4}$$

where q_k is the probability that node k may be activated, and λ_k is the data packet arrival rate at link k.

5.3 Simulation Results

The behaviors of HAMA is simulated in two scenarios: fully connected networks with different numbers of nodes, and multihop networks with different radio transmission ranges. The delay and throughput attributes of HAMA are gathered in each simulation, and compared with those of NAMA [1] and UxDMA [12] in the same simulation scenarios.

In the simulations, we do not model the bandwidth of the radio channel with specific numbers, but use more abstract terms, such as *packets per time slot* for both arrival rate and throughput, which can be later translated into common bandwidth metrics, such as *Mbps* (megabits per second), given certain packet size distribution and transmission media. The following parameters and behaviors are assumed in the simulations:

- The network topologies are static to evaluate the scheduling performance of the algorithms, only.
- Signal propagation in the channel follows the free-space model and the effective range of radio is determined by the power level of the radio. Radiation energy outside the effective transmission range of the radio is considered negligible interference to other communications. All radios have the same transmission range.
- 30 pseudo-noise codes are available for code assignments, i.e., $|C_{pn}| = 30$.
- Packets are served in First-In First-Out (FIFO) order. Only one packet can be transmitted in a time slot.
- All nodes have the same broadcast packet arrival rate for all protocols (HAMA, NAMA and UxDMA). In addition, HAMA is loaded with the same amount of unicast traffic as broadcast traffic to manifest the unicast capability of HAMA. The overall load for HAMA is thus twice as much as that of NAMA and UxDMA. The destinations of the unicast packets in HAMA are evenly distributed on all outgoing links.

- The duration of the simulation is 100,000 time slots, long enough to collect the metrics of interests.

In UxDMA, a constraint set is derived for broadcast activations as NAMA does, which is give by UxDMA-NAMA = $\{V_{tr}^{0}, V_{tt}^{1}\}$. The notation of each symbol is referred to the original paper in [12]. Constraint V_{tr}^{0} forbids a node from transmitting and receiving at the same time, while V_{tt}^{1} eliminates hidden terminal problem.

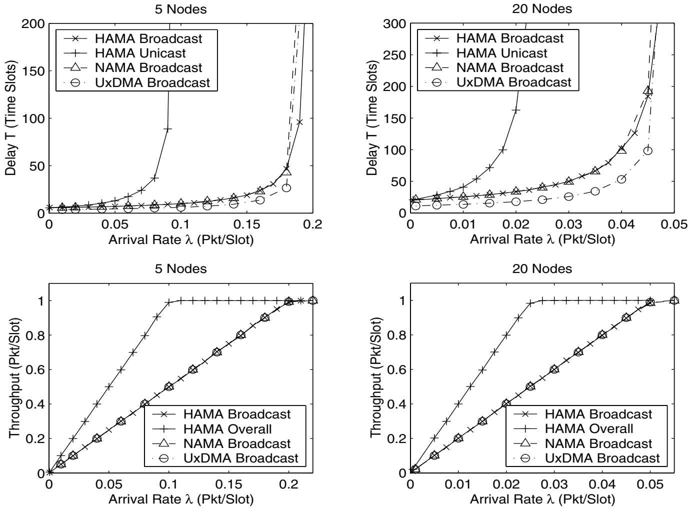


Fig. 4. Average Packet Delays In Fully-Connected Networks

In the fully connected scenarios, simulations were carried out in two configurations: 5- and 20-node networks, to manifest the effects of different contention levels. Fig. 4 shows the average delay values on the first row and average throughput on the second for HAMA, NAMA and UxDMA-NAMA, respectively, under different loads on each node in the two configurations. The horizontal parts in the throughput plots indicate the total network capacities provided by different protocols. Since all nodes are within one hop to each other, there can be only one unicast or broadcast in each time slot. The network throughput tops when the loads sum up to one.

In the multihop scenario, the simulations were conducted in networks that are generated by randomly placing 100 nodes within an area of 1000×1000 square meters. To simulate infinite plane that has constant node placement density, the opposite sides of the square are seamed together, which visually turns the square area into a torus. The power of the transceiver on each node was set to 100, 200, and 300 meters, respectively, so that the network topology and contention levels in these simulations varied accordingly.

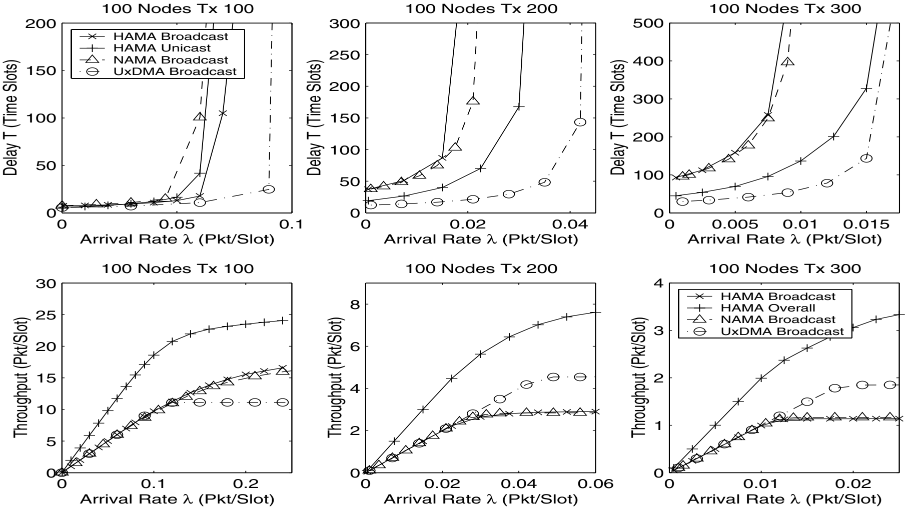


Fig. 5. Average Packet Delays In Multihop Networks

Fig. 5 shows the delay and throughput performance of HAMA, NAMA and UxDMA-NAMA in multihop networks. UxDMA-NAMA is better than HAMA and NAMA at broadcasting in some of the multihop networks, owing to its global knowledge about topologies. However, HAMA outperforms UxDMA-NAMA in overall network throughput.

Overall, HAMA has achieved much better performance than a previously proposed protocol, NAMA [1], by requiring only a little more processing on the neighbor information. Comparing HAMA with UxDMA, which uses global topology information, HAMA sustains similar broadcasting throughput, in addition to the extra opportunities for sending unicast traffic. The dependence on only two-hop neighbor information is also a big advantage over UxDMA.

6 Conclusion

We have introduced HAMA, a new distributed channel access scheduling protocol that dynamically determines the node activation schedule for both broadcast and unicast traffics. HAMA only requires two-hop neighborhood information, and avoids the complexities of prior collision-free scheduling approaches that demand global topology information. We have also analyzed the per-node delay and per-system throughput attributes of HAMA and NAMA [1], a node activation protocol, and compared system performance of HAMA with that of NAMA and UxDMA [12] by simulation.

References

1. L. Bao and J.J. Garcia-Luna-Aceves. A New Approach to Channel Access Scheduling for Ad Hoc Networks. In *Proc. ACM Seventh Annual International Conference on Mobile Computing and networking*, Rome, Italy, Jul. 16-21 2001.
2. D. Bertsekas and R. Gallager. *Data Networks, 2nd edition.* Prentice Hall, Englewood Cliffs, NJ, 1992.
3. A. Ephremides and T.V. Truong. Scheduling broadcasts in multihop radio networks. *IEEE Transactions on Communications*, 38(4):456–60, Apr. 1990.
4. S. Even, O. Goldreich, S. Moran, and P. Tong. On the NP-completeness of certain network testing problems. *Networks*, 14(1):1–24, Mar. 1984.
5. C.L. Fullmer and J.J. Garcia-Luna-Aceves. Floor acquisition multiple access (FAMA) for packet-radio networks. In *ACM SIGCOMM '95*, pages 262–73, Cambridge, MA, USA, Aug. 28 -Sep. 1 1995.
6. J.J. Garcia-Luna-Aceves and J. Raju. Distributed assignment of codes for multihop packet-radio networks. In *MILCOM 97 Proceedings*, pages 450–4, Monterey, CA, USA, Nov. 2-5 1997.
7. M. Joa-Ng and I.T. Lu. Spread spectrum medium access protocol with collision avoidance in mobile ad-hoc wireless network. In *IEEE INFOCOM '99*, pages 776–83, New York, NY, USA, Mar. 21-25 1999.
8. P. Karn. MACA - a new channel access method for packet radio. In *Proceedings ARRL/CRRL Amateur Radio 9th Computer Networking Conference*, New York, Apr. 1990.
9. L. Kleinrock and F.A. Tobagi. Packet switching in radio channels. I. Carrier sense multiple-access modes and their throughput-delay characteristics. *IEEE Transactions on Communications*, COM-23(12):1400–16, Dec 1975.
10. L. Lamport. Time, clocks, and the ordering of events in a distributed system. *Communications of the ACM*, 21(7):558–65, Jul. 1978.
11. T. Makansi. Trasmitter-Oriented Code Assignment for Multihop Radio Net-works. *IEEE Transactions on Communications*, 35(12):1379–82, Dec. 1987.
12. S. Ramanathan. A unified framework and algorithm for channel assignment in wireless networks. *Wireless Networks*, 5(2):81–94, 1999.
13. R. Ramaswami and K.K. Parhi. Distributed scheduling of broadcasts in a radio network. In *IEEE INFOCOM'89*, volume 2, pages 497–504, Ottawa, Ont., Canada, Apr. 23-27 1989. IEEE Comput. Soc. Press.

Towards Efficient Decision Rules for Admission Control Based on the Many Sources Asymptotics

Gergely Seres[1], Árpád Szlávik[1], János Zátonyi[2], and József Bíró[2]

[1] Traffic Lab, Ericsson Research Hungary, P.O. Box 107, H-1300 Budapest, Hungary,
Gergely.Seres@eth.ericsson.se
[2] HSN Lab, DTT, Budapest University of Technology and Economics, P.O. Box 91,
H-1521 Budapest, Hungary

Abstract. This paper introduces new admission criteria that enable the use of algorithms based on the many sources asymptotics in real-life applications. This is achieved by a significant reduction in the computational requirements and by moving the computationally intensive tasks away from the timing-sensitive decision instant. It is shown that the traditional overflow-probability type admission control method can be reformulated into a bandwidth-requirement type and a buffer-requirement type methods and that these methods are equivalent when used for admission control. The original and the two proposed methods are compared through the example of fractional Brownian motion traffic.

1 Introduction

Bandwidth requirement estimation is a key function in networks intending to provide quality of service (QoS) to their users. Network devices in QoS-capable networks must be able to control the amount of traffic they handle. This is generally performed by using some form of admission control. There are two commonly used methods for determining whether a new connection can be allowed to enter the system: in the first one an estimate of the buffer overflow probability is computed based on the properties of the new and the already active flows in the system, while the second method computes the bandwidth requirement of the existing traffic flows. When using the first method for admission control decisions, the devices check the computed overflow probability against the target overflow probability. If the second method is used, the bandwidth requirement of the existing flows is increased by the predicted bandwidth usage of the new flow and the result is compared to the capacity of the system.

Often, the second method is preferred over the first, mainly because it results in a quantity – the bandwidth requirement – that is more tractable and more useful than the estimate of the overflow probability. The on-line estimation of the bandwidth requirement of the traffic enables the network operator to track the amount of allocated (and free) capacity in the network. Furthermore, the impact of network management actions (e.g. directing more traffic on the link) on the resource status of the network can be more easily assessed. The overflow probability on the other hand is a less straightforward quantity that depends on the

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 166–177, 2002.

parameters of the queueing system in a more complex way, thus changes in them imply a less tractable and computationally more complex update procedure.

Accordingly, most of the work to date has focused on algorithms that quantify the bandwidth requirement of traffic flows. The most widespread approaches are based on the notion of the effective bandwidth, a comprehensive review of which is given in [5]. A group of algorithms use the Chernoff bound or the Hoeffding bound to derive simplified and directly applicable formulae for the effective bandwidth in case of bufferless statistical multiplexing [9]. For buffered resources, the theory of large deviations was shown to be a very capable method for calculating the bandwidth requirement of traffic flows. There are two asymptotics that can be used for this purpose: the large buffer asymptotics and the many sources asymptotics. The large buffer asymptotics provide a rate function describing the decay rate of the tail of the probability of buffer overflow when the size of the buffer gets very large. The many sources asymptotics also offer a rate function but with the assumption that the number of traffic flows in the system gets very large, while the traffic mix, per-source buffer space and system per-source capacity are held constant. Both asymptotics discussed so far provide an overflow-probability type quantity.

Using the large buffer asymptotics it is easy to switch from the overflow probability representation to the bandwidth requirement representation. However, algorithms relying on this asymptotics [6] do not account for the gain arising from the statistical multiplexing of many traffic flows. In recent years, the second asymptotic regime, the many sources asymptotics (and its Bahadur-Rao improvement) have been described and investigated in [3], [2], [1] and [7]. In the native form, the many sources asymptotics provide a rate function that can be used to estimate the probability of overflow. The computation of this rate function involves two optimisations in two variables. Yet, if it is the bandwidth requirement that is of interest, another optimisation has to be performed that requires the recomputation of the two original optimisations in each step. Despite of its complexity, this bandwidth requirement estimate is appealing because it incorporates the statistical properties of the traffic along with its QoS requirements and it also embraces the statistical multiplexing gain that occurs on the multiplexing link. However, the use of this estimator in real-time applications is not feasible because of its computational complexity.

This paper introduces a new method for computing the bandwidth requirement of traffic flows that is based on the many sources asymptotics as well. Instead of the three embedded optimisations that previous approaches required, it comprises only of two optimisations that directly result in an estimate of the bandwidth requirement. The method is favourable to on-line measurement-based application, since the admission decision step is simplified and the more involving computations can be done in the background. It is shown that the new and the old methods for obtaining the bandwidth requirement are equivalent.

The rest of this paper is organised as follows. Section 2 presents a brief overview of the many sources asymptotics and describes three admission decision methods based directly on this asymptotics. The proposed computationally more favourable method for computing the bandwidth requirement is introduced in

Sect. 3, and the equivalence is proven. The operation of the novel method is demonstrated with the example of fractional Brownian motion traffic in Sect. 4. Conclusions are given in Sect. 5. The Appendix shows that similar results can be achieved for computing the buffer requirement of traffic flows.

2 Overflow-Probability Based Admission Criteria

This section presents an overview on the many sources asymptotics. Next, a collection of admission control methods are reviewed, all of which build on the asymptotic property of the overflow probability.

2.1 Many Sources Asymptotics

The asymptotic regime described by the many sources asymptotics can be used to form an estimate of the probability of buffer overflow in the system as follows. Let us consider a buffered communication link with transmission capacity C, buffer size B, which carries N independent flows multiplexed in the system. N is viewed as a scaling factor, i.e. we can identify a per-source transmission capacity $c = C/N$ and a per-source buffer size $b = B/N$. Further, let the stochastic process $X[0,t)$ denote the total amount of work arriving at the system during the time interval $[0,t)$. Let us assume that $X[0,t)$ has stationary increments.

Conclusions on the behaviour of this system can be derived by investigating a queueing system of infinite buffer size that is served by a finite capacity server with service rate $C = cN$. In order to account for the finite buffer size $B = bN$ of the real system, the probability of buffer overflow in the original system can be deduced from the proportion of time over which the queue length, $Q(C,N)$, is above the finite level B. In this system, where the system parameters (cN, bN) and the workload $(X[0,t))$ are scaled by the number of sources, an asymptotic equality can be obtained in N for the probability of overflow:

$$\lim_{N\to\infty} \frac{1}{N} \log P\{Q(cN,N) > bN\} = \sup_{t>0} \inf_{s>0} \left\{ st\, \frac{\alpha(s,t)}{N} - s(b+ct) \right\} \stackrel{\text{def}}{=} -I \ . \tag{1}$$

Here $\alpha(s,t)$ (the so-called effective bandwidth [5]) is defined as

$$\alpha(s,t) \stackrel{\text{def}}{=} \frac{1}{st} \log E\left[e^{sX[0,t)}\right] \tag{2}$$

and I is called the asymptotic rate function, which depends on the per-source system parameters and on the scaled workload process. This result was proven for discrete time in [2] and for continuous time in [3]. Equation (2) practically means that for N large, the probability of overflow can be approximated as $P\{Q(C,N) > B\} \approx e^{-NI}$, where $-NI$ can be computed from (1) as

$$-NI = \sup_{t>0} \inf_{s>0} \{st\, \alpha(s,t) - s(B+Ct)\} \ . \tag{3}$$

The approximation above can also be reasoned in a less rigorous, but brief and intuitive manner as follows [7]. The Chernoff bound can be used to approximate the probability that the workload $X[0,t)$ exceeds Ct, the offered service in $[0,t)$ and in addition it fills up the buffer space B: $P\{X[0,t) > B + Ct\} \approx \inf_{s>0} \exp\{st\,\alpha(s,t) - s(B+Ct)\}$. The steady state queue length distribution can be described by $Q = \sup_{t>0}\{X[0,t) - Ct\}$ provided that the $X[0,t)$ process has stationary increments. This way, the probability of the queue length exceeding the buffer level B is $P\{Q > B\} \approx P\{\sup_{t>0}\{X[0,t) - Ct\} > B\} \approx \sup_{t>0} P\{X[0,t) > B + Ct\} \approx e^{-NI}$.

For the sake of simplifying further discussions, let us define the function $J(s,t) \overset{\text{def}}{=} st\,\alpha(s,t) - s(B+Ct)$. In (3), the evaluation of $\sup_{t>0}\inf_{s>0} J(s,t)$ is computationally complex as a double optimisation has to be performed after the computation or estimation of the effective bandwidth of $X[0,t)$. Since the optimisations are embedded, first the optimal (minimal) s has to be found which still depends on t. Placing this optimal s into $J(s,t)$, the task is its maximisation with respect to t. For a more formal and concise discussion the following notation is introduced:

$$s^*(t) \overset{\text{def}}{=} \arg\inf_{s>0} J(s,t), \quad t^* \overset{\text{def}}{=} \arg\sup_{t>0} J(s^*(t),t) \quad \text{and} \quad s^* \overset{\text{def}}{=} s^*(t^*) \ . \tag{4}$$

Now, the extremising pair of $J(s,t)$ is (s^*,t^*) and thus $-NI = J(s^*,t^*)$. The extremising values t^* and s^* are commonly termed as the critical time and space scales, respectively. The intuitive explanation of the critical time scale is that it is the most probable time interval after which overflows occur in the multiplexing system (i.e. the most likely length of the busy period prior to overflow). Although many other busy periods may contribute to the total overflow, large deviation theory takes into account only the most probable one, which is the most dominant in the asymptotic sense. The rationale behind the critical space parameter is that it captures the statistical behaviour of the workload process, that is the amount of achievable statistical multiplexing gain and the burstiness. Critical space values close to 0 describe a source (or an aggregate) that can benefit from statistical multiplexing, while larger values infer a higher bandwidth requirement. Finally, it is also worth noting that s^* and t^* always depend on the system parameters C, B and the statistical properties of $X[0,t)$.

In practical applications there is a QoS requirement, which is often specified as a constraint for the probability of buffer overflow ($e^{-\gamma}$). In order to admit a source the following criterion has to be satisfied:

$$P\{Q(C,N) > B\} \approx e^{-NI} \leq e^{-\gamma} \quad \text{or} \quad \sup_{t>0}\inf_{s>0} J(s,t) \leq -\gamma \ . \tag{5}$$

2.2 Equivalent Admission Criteria

The inequalities in (5) define an admission rule that uses the method of the many sources asymptotics in the native form. In this original form, the probability of buffer overflow is estimated using $X[0,t)$, B and C as the input quantities, whilst the target overflow probability is used as the performance criterion.

It is possible to set up two other criteria that can be used for admission control decisions. As it was mentioned in the introduction, it is often preferable to express the bandwidth requirement of the traffic and compare this quantity to the server capacity. In order to form an estimate of the bandwidth requirement of the traffic, another optimisation has to be performed. For this, the server capacity has to be treated as a free variable and given the workload process, the buffer size and the QoS requirement, the smallest server capacity has to be identified for which the system still satisfies the performance criterion put forward in (5). The resulting quantity

$$C_{\mathrm{equ}} \stackrel{\mathrm{def}}{=} \inf \left\{ C : \sup_{t>0} \inf_{s>0} J(s,t) \le -\gamma \right\} \tag{6}$$

is termed in the rest of the paper as the equivalent capacity.[1] Then the admission criterion can be written as

$$C_{\mathrm{equ}} \le C \ . \tag{7}$$

A similar, but less frequently used criterion can be defined that allows admission decisions to be made based on the available buffer space. In this case, the buffer requirement of the traffic is determined using a similar triple optimisation as in (6), but this time taking $X[0,t)$, C and the QoS requirement as the input quantities and B as the performance constraint:

$$B_{\mathrm{req}} \stackrel{\mathrm{def}}{=} \inf \left\{ B : \sup_{t>0} \inf_{s>0} J(s,t) \le -\gamma \right\} \quad \text{and} \quad B_{\mathrm{req}} \le B \ . \tag{8}$$

Figure 1 presents a summary of the three methods with respect to their input parameters and the quantity they use as a constraint in the decision criterion. The methods are equivalent in the sense that in a given context they arrive at the same decision. When it comes to numerical evaluation, the first (original) method with the double optimisation is, however, significantly less demanding than the others involving three embedded optimisations.

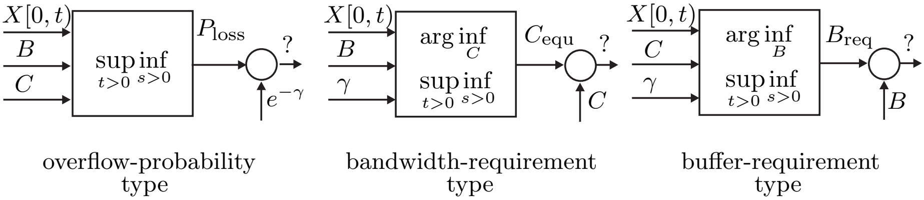


Fig. 1. Admission decision methods

[1] Following the terminology of previous works, the term effective bandwidth is reserved for $\alpha(s,t)$, which is not directly associated with the minimal service rate required to meet the QoS target.

3 The Improved Bandwidth Requirement Estimator

This section introduces an alternative method for computing the equivalent capacity. The advantage of this new method is that its computational complexity is reduced to a double optimisation, resulting in a similar formula to the one used in the rate-function based estimation of the buffer overflow probability. It is shown that the estimation of the equivalent capacity using the proposed method arrives at the same decision as the method in (6) and (7). The proposed method infer a new optimisation function resulting in an alternative set of space and time scales. The equivalence of the respective methods for estimating the buffer requirement can be proven in an identical manner (see the Appendix).

3.1 Alternative Definition of the Equivalent Capacity

Let us introduce $K(s,t)$ as

$$K(s,t) \overset{\text{def}}{=} \alpha(s,t) + \frac{\gamma}{st} - \frac{B}{t} \ , \tag{9}$$

which is obtained from the isolation of C from $J(s,t) = -\gamma$. Namely, $K(s,t) = C$ holds after the rearrangement.[2] By defining a new double optimisation

$$\widetilde{C}_{\text{equ}} \overset{\text{def}}{=} \sup_{t>0} \inf_{s>0} K(s,t) \ , \tag{10}$$

similarly to (3), the extremisers are attained in the form of

$$s^\dagger(t) \overset{\text{def}}{=} \arg\inf_{s>0} K(s,t), \quad t^\dagger \overset{\text{def}}{=} \arg\sup_{t>0} K(s^\dagger(t),t) \quad \text{and} \quad s^\dagger \overset{\text{def}}{=} s^\dagger(t^\dagger) \ . \tag{11}$$

The extremising pair of the double optimisation in (10) is then $(s^\dagger, t^\dagger)$ and these are the alternative space and time scales, respectively.

It can be proven that $\widetilde{C}_{\text{equ}} = C_{\text{equ}}$ holds. In other words, we need only two optimisations instead of three to arrive at the equivalent capacity C_{equ}. This is shown in the next subsection using the subsequent theorem.

Theorem 1. *The following two strict inequalities are equivalent:*

$$J(s^*,t^*) < -\gamma \Longleftrightarrow K(s^\dagger,t^\dagger) < C \ , \tag{12}$$

furthermore the equations

$$J(s^*,t^*) = -\gamma \ , \tag{13}$$

$$K(s^\dagger,t^\dagger) = C \tag{14}$$

[2] The so-called Bahadur-Rao improvement, as described in [1], introduces a prefactor to the estimate of the overflow probability in (5). For the improvement of the equivalent capacity (and the buffer requirement) this manifests in a modified QoS constraint for (5) in the following form: $\gamma' = \gamma - \frac{\frac{1}{2}\log(4\pi\gamma)}{1+\frac{1}{2\gamma}}$.

are equivalent as well and consequently the two strict inequalities below also imply each other:

$$J(s^*,t^*) > -\gamma \ , \tag{15}$$

$$K(s^\dagger,t^\dagger) > C \ . \tag{16}$$

Proof. First of all, note that any two of the three equivalences, (12), (13)⇔(14) and (15)⇔(16), imply the third one, therefore it is enough to prove the second and the third assertion only. The proof of the statements of Theorem 1 uses three lemmas.

Lemma 1. *For all* $t > 0$*, (17) ⇔ (18):*

$$J(s^*(t),t) < -\gamma \ , \tag{17}$$

$$K(s^\dagger(t),t) < C \ . \tag{18}$$

Proof. Suppose (17) holds. The isolation of C gives $K(s^*(t),t) \overset{\text{by rearr. (17)}}{<} C$. Then, the definition of $s^\dagger(t)$ can be used to obtain $K(s^\dagger(t),t) \overset{\text{by def. of } s^\dagger(t)}{\leq} K(s^*(t),t)$. These two inequalities together entail that (18) holds as well. In the other direction, if (18) holds, $J(s^*(t),t) \overset{\text{by def. of } s^*(t)}{\leq} J(s^\dagger(t),t) \overset{\text{by rearr. (18)}}{<} -\gamma$, consequently (18)⇒(17) as well and thus Lemma 1 is proven. □

Lemma 2. *For all* $t > 0$*, (19) ⇔ (20):*

$$J(s^*(t),t) = -\gamma \ , \tag{19}$$

$$K(s^\dagger(t),t) = C \ . \tag{20}$$

Proof. If (19) holds, then $K(s^\dagger(t),t) \geq C$ because otherwise (17) should hold by the assertion of Lemma 1 contradicting (19). This means that $C \overset{\text{by L1 and (19)}}{\leq} K(s^\dagger(t),t) \overset{\text{by def. of } s^\dagger(t)}{\leq} K(s^*(t),t) \overset{\text{by rearr. (19)}}{=} C$, therefore equality has to hold throughout this chain of inequalities, consequently (20) holds. Now suppose that (20) holds. Then $-\gamma \overset{\text{by L1 and (20)}}{\leq} J(s^*(t),t) \overset{\text{by def. of } s^*(t)}{\leq} J(s^\dagger(t),t) \overset{\text{by rearr. (20)}}{=} -\gamma$, so (20)⇒(19) as well and hence Lemma 2 is proven. □

Lemma 3. *For all* $t > 0$

$$J(s^*(t),t) > -\gamma \Longleftrightarrow K(s^\dagger(t),t) > C \ . \tag{21}$$

Proof. Lemma 3 is a straightforward consequence of Lemma 1 and Lemma 2. □

Continuing with the proof of Theorem 1, let us suppose that (13) holds. Substituting t^* in Lemma 2 implies that $K(s^\dagger(t^*),t^*) = C$. On the other hand, $K(s^\dagger,t^\dagger) = K(s^\dagger(t^\dagger),t^\dagger) \geq K(s^\dagger(t^*),t^*)$ by the definition of $t^\dagger$ (note that in general $s^\dagger$ can not be used instead of $s^\dagger(t^*)$). Thus $K(s^\dagger(t^\dagger),t^\dagger) \overset{\text{by def. of } t^\dagger}{\geq}$

$K(s^\dagger(t^*),t^*) \overset{\text{by L2 with } t=t^*}{=} C$. By Lemma 2 and Lemma 3 with $t = t^\dagger$ this inequality implies that $J(s^*(t^\dagger),t^\dagger) \geq -\gamma$. Hence $-\gamma \overset{(13)}{=} J(s^*(t^*),t^*) \overset{\text{by def. of } t^*}{\geq} J(s^*(t^\dagger),t^\dagger) \overset{\text{prev., L2 and L3 with } t=t^\dagger}{\geq} -\gamma$, from which it can be concluded that equality holds along the chain of inequalities, accordingly $J(s^*(t^\dagger),t^\dagger) = -\gamma$. Lemma 2 with $t = t^\dagger$ then implies that (14) holds. Vice versa, if (14) holds, then $J(s^*,t^*) = J(s^*(t^*),t^*) \overset{\text{by def. of } t^*}{\geq} J(s^*(t^\dagger),t^\dagger) \overset{\text{by L2 with } t=t^\dagger}{=} -\gamma$, thereupon $C \overset{(14)}{=} K(s^\dagger(t^\dagger),t^\dagger) \overset{\text{by def. of } t^\dagger}{\geq} K(s^\dagger(t^*),t^*) \overset{\text{prev., L2 and L3 with } t=t^*}{\geq} C$. The consequence is that $K(s^\dagger(t^*),t^*) = C$ holds and by Lemma 2 with $t = t^*$ it turns out that (13) holds as well. Thus the proof of equivalence (13)⇔(14) is done. For the proof of the third statement of Theorem 1, first suppose (15) holds. Then $K(s^\dagger,t^\dagger) = K(s^\dagger(t^\dagger),t^\dagger) \overset{\text{by def. of } t^\dagger}{\geq} K(s^\dagger(t^*),t^*) \overset{\text{by L3 with } t=t^*}{>} C$, consequently (15)⇒(16). Finally, supposing (16) holds, $J(s^*,t^*) = J(s^*(t^*),t^*) \overset{\text{by def. of } t^*}{\geq} J(s^*(t^\dagger),t^\dagger) \overset{\text{by L3 with } t=t^\dagger}{>} -\gamma$, subsequently (16)⇒(15) as well. Therefore the equivalence (15)⇔(16) is proven as well. Equivalence (12) follows from the other two equivalences, hence the proof of Theorem 1 is completed. □

3.2 Equivalence of the Two Definitions of the Equivalent Capacity

Theorem 1 can now be used to prove that the equivalent capacities defined by (6) and (10) are equal.

Corollary 1. *The equivalent capacity defined by the double optimisation in (10) equals the one defined by the triple optimisation in (6):* $K(s^\dagger,t^\dagger) = \widetilde{C}_{\text{equ}} = C_{\text{equ}}$.

Proof. Observe that $J(s,t)$ is (strictly) monotonously decreasing in the variable C for fixed B and γ. It is easy to see that $C_{\text{equ}} = \inf\{C : \sup_{t>0}\inf_{s>0} J(s,t) \leq -\gamma\} \overset{\text{str. mon. decr.}}{=} \inf\{C : \sup_{t>0}\inf_{s>0} J(s,t) = -\gamma\}$. The consequence of this is that $J((s^*(C_{\text{equ}}),t^*(C_{\text{equ}})) = -\gamma$ (note that the extremisers depend on the variables C and B in this case, but B is fixed here, therefore only the dependence on variable C is indicated). By Theorem 1 it follows that $K(s^\dagger,t^\dagger) = C_{\text{equ}}$ as well (here the optimising parameters depend on variables B and γ, but those are fixed). However, this is exactly the definition of $\widetilde{C}_{\text{equ}}$ and that corresponds to the assertion. □

The respective optimiser pairs $(s^*(B,C),t^*(B,C))$ and $(s^\dagger(B,\gamma),t^\dagger(B,\gamma))$ do not coincide in general, they are not even comparable as such, since they depend on a different set of variables. Nevertheless, on the boundary of the acceptance region ($J(s^*,t^*) = -\gamma \Leftrightarrow (C_{\text{equ}} =)K(s^\dagger,t^\dagger) = C$) the same parameter values are the optimisers of the two problems:

Proposition 1. *If one of the double optimisations (3) and (10) has a unique extremising pair and* $J(s^*,t^*) = -\gamma$ *(or* $K(s^\dagger,t^\dagger) = C$*), then the two extremiser pairs coincide,* $t^* = t^\dagger$ *and* $s^* = s^\dagger$.

Proof. Assume $(s^\dagger, t^\dagger)$ is unique. By supposing any of the two (equivalent) equalities as the second assumption, $J(s^*(t^*), t^*) = -\gamma$ and $K(s^\dagger(t^\dagger), t^\dagger) = C$ hold. By Lemma 2 with $t = t^*$ this means that $K(s^\dagger(t^*), t^*) = C$ holds as well. Since $K(s^\dagger(t^\dagger), t^\dagger) = C$, then $t^* = t^\dagger$ by the uniqueness property. On the other hand, by rearranging $K(s^\dagger(t^*), t^*) = C$, $J(s^\dagger(t^*), t^*) \overset{\text{rearr.}}{=} -\gamma = J(s^*(t^*), t^*)$ is obtained. This proves $s^* = s^*(t^*) \overset{\text{uniq.}}{=} s^\dagger(t^*) \overset{t^*=t^\dagger}{=} s^\dagger$. The proof of the statement is almost the same when assuming the uniqueness of (s^*, t^*). □

4 Comparison of the Methods for fBm Traffic

This section presents a comparison of the three admission control methods discussed in Sect. 2.2 using the new formulae developed in Sect. 3.1 and in the Appendix. The traffic case used is fractional Brownian motion (fBm), which involves closed-form formulae due to its Gaussian nature.

4.1 Key Formulae for Fractional Brownian Motion Traffic

The stochastic process $\{Z_t, t \in \mathbb{R}\}$ is called normalized fractional Brownian motion with self-similarity (Hurst-) parameter $H \in (0,1)$ if it has stationary increments and continuous paths, $Z_0 = 0$, $E[Z_t] = 0$, $Var[Z_t] = |t|^{2H}$ and if Z_t is a Gaussian process. Let us define the process $X[0,t) \overset{\text{def}}{=} mt + Z_t$, for $t > 0$. It is known as fractional Brownian traffic and can be interpreted as the amount of traffic offered to the multiplexer in the time interval $[0,t)$. This is a so-called self-similar model, which has been suggested for the description of Internet traffic aggregates [8], [4].

Using this model the effective bandwidth (2) can be written as $\alpha(s,t) = m + \frac{s\sigma^2 t^{2H-1}}{2}$ and accordingly $J(s,t) = st\,m + \frac{s^2\sigma^2 t^{2H}}{2} - s(B+Ct)$. The extremisers for $J(s,t)$ and $-NI$ can be found in Table 1[3], where $\kappa(H) \overset{\text{def}}{=} H^H(1-H)^{1-H}$.

The equivalent capacity can be evaluated in two ways, either using the definition in (6) or the method proposed in this paper (10). $\widetilde{C}_{\text{equ}}$ requires the direct evaluation of $K(s,t)$ (9) at the alternative critical space and time scales $(s^\dagger, t^\dagger)$ (11), i.e. "only" a double optimisation is necessary. For fBm traffic $K(s,t) = m + \frac{1}{2}\sigma^2 s t^{2H-1} + \frac{\gamma}{st} - \frac{B}{t}$, its extremisers and $\widetilde{C}_{\text{equ}}$ are listed in Table 1. If C_{equ} is calculated in the conventional way (6), the third optimisation (with respect to C) can be exchanged for solving $-NI = -\gamma$ for $C = C_{\text{equ}}$ (as seen in the proof of Corollary 1).[4] It can be checked that $C_{\text{equ}} = \widetilde{C}_{\text{equ}}$ as expected (using the definition of $\kappa(H)$). In a similar way, s', t' and $\widetilde{B}_{\text{req}}$ (see the Appendix) can be computed (see Table 1) and it also turns out that $\widetilde{B}_{\text{req}} = B_{\text{req}}$.

[3] An identical expression for the approximation of the overflow probability was obtained in [8] with a different approach.

[4] This simplification can be done only because $-NI$ is an explicit function of C in the fBm case (C can be isolated from the equation). In most other cases the third optimisation must be done through several double optimisations of $J(s,t)$ for different values of C in order to locate $C = C_{\text{equ}}$ for which $-NI = -\gamma$.

Table 1. Comparison of the three admission control methods for fBm traffic

	$f(s,t)$		
	$J(s,t)$	$K(s,t)$	$L(s,t)$
$s^{\text{opt}}(t) =$ $\arg\inf\limits_{s>0} f(s,t)$	$s^*(t) =$ $\frac{t^{-2H}(B+(C-m)t)}{\sigma^2}$	$s^\dagger(t) =$ $\frac{\sqrt{2\gamma}t^{-H}}{\sigma}$	$s'(t) =$ $\frac{\sqrt{2\gamma}t^{-H}}{\sigma}$
$t^{\text{opt}} =$ $\arg\sup\limits_{t>0} f(s^{\text{opt}}(t),t)$	$t^* =$ $\frac{H}{1-H}\frac{B}{C-m}$	$t^\dagger =$ $2^{-\frac{1}{2H}}\left(\frac{B}{(1-H)\sqrt{\gamma}\sigma}\right)^{\frac{1}{H}}$	$t' =$ $\left(\frac{H\sqrt{2\gamma}\sigma}{C-m}\right)^{\frac{1}{1-H}}$
$s^{\text{opt}} = s^{\text{opt}}(t^{\text{opt}})$	$s^* = \frac{1-H}{\kappa(H)^2}\cdot$ $\frac{(C-m)^{2H}B^{1-2H}}{\sigma^2}$	$s^\dagger =$ $\frac{2(1-H)\gamma}{B}$	$s' = \left(\frac{C-m}{H}\right)^{\frac{H}{1-H}}\cdot$ $\left(\sqrt{2\gamma}\sigma\right)^{\frac{1}{1-H}}2\gamma$
$\sup\limits_{t>0}\inf\limits_{s>0} f(s,t)$	$-NI =$ $-\frac{(C-m)^{2H}B^{2-2H}}{2\kappa(H)^2\sigma^2}$	$\widetilde{C}_{\text{equ}} = m + H\cdot$ $\left(2\gamma\sigma^2\right)^{\frac{1}{2H}}\left(\frac{1-H}{B}\right)^{\frac{1-H}{H}}$	$\widetilde{B}_{\text{req}} = \left(\frac{H}{C-m}\right)^{\frac{H}{1-H}}\cdot$ $(1-H)\left(\sqrt{2\gamma}\sigma\right)^{\frac{1}{1-H}}$

Confirming the statements in the previous section, it is apparent from Table 1 that the critical space scales s^*, $s^\dagger$ and s' are usually different and depend on different parameter sets. For given B, C, γ, m, H and σ, the corresponding scales match only when equalities (13) and (14) hold. An interesting consequence of this fact is that the solution of $t^*(B,C,m,H) = t^\dagger(B,\gamma,\sigma,H)$ for C results in the equivalent capacity C_{equ} and its solution for γ is NI. Similar statements are valid for the space scales as well.

4.2 A Numerical Example

In this subsection a numerical example is presented to demonstrate the results of the previous subsection. Let us take the fBm model of one of the Bellcore Ethernet data traces [4]: $m_1 = 138\,135\,\text{byte/s}$, $\sigma_1 = 89\,668\,\text{byte/s}^H$, $H = 0.81$. Assume that $N = 100$ of such sources are multiplexed into a buffer. Hence, the model parameters of the fBm model for the aggregate traffic workload become: $m = 13.8135\,\text{Mbyte/s}$, $\sigma = 0.89668\,\text{Mbyte/s}^H$, $H = 0.81$. The buffer size is chosen to be $B = 5.3\,\text{Mbyte}$, the service rate is $C = 16\,\text{Mbyte/s}$ and let the constraint for the overflow be $e^{-16} \approx 10^{-7}$ ($\gamma = 16$). For these system parameters, the extremiser pair is $(s^*,t^*) = (1.453, 7.091)$ and therefore $-NI = -20.26$. Clearly, $-NI < -\gamma$, i.e. the QoS requirement is fulfilled. The alternative critical scales and the equivalent capacity are obtained as $(s^\dagger,t^\dagger) = (1.147, 8.203) \neq (s^*,t^*)$ and $C_{\text{equ}} = \widetilde{C}_{\text{equ}} = 15.568\,\text{Mbyte/s}$, thus $C_{\text{equ}} < C$ holds and there is $0.432\,\text{Mbyte/s}$ of free service capacity.

5 Conclusion

This paper has introduced a new method for the computation of the equivalent capacity (and the buffer requirement) of traffic flows that is based on the many sources asymptotics. In contrast to the method directly building on the asymptotic rate function, the new method involves only two embedded optimisations instead of three, thus it significantly reduces the computational complexity of the task. It has been shown that the two methods are equivalent.

The presented method of deriving the equivalent capacity leads to an alternative domain of time and space scales. In a given system the optimisation defininig the equivalent capacity estimate $(C_{\text{equ}} =) \widetilde{C}_{\text{equ}}$ (10) yields different optimal parameter values than that defining the estimate of the overflow probability e^{-NI} (3). Consequently, the substitution of the extremisers of $J(s,t)$ into $K(s,t)$ (9) does not lead to a correct estimate of the equivalent capacity. The only exception is the boundary of the admission region, where the two extremising pairs coincide.

In terms of applicability, it can be shown that the method of the equivalent capacity computation is more appropriate for real-time operation than those based on the asymptotic rate function, especially if the workload process is measured on-line (measurement-based admission control). Recall the admission methods defined by (5) and (6), (7). In practice these admission rules are performed at the arrival of a new flow. The effective bandwidth estimate has to be adjusted in order to take the new flow into account. For example, let us assume that the new flow is described by its peak rate only. Then $\alpha^+(s,t) = \alpha(s,t) + p$ is a conservative adjustment. With the rate-function based admission method, the double optimisation has to be re-evaluated in order to update the estimate of the overflow probability: $-NI^+ = \sup_{t>0} \inf_{s>0} \{st\,\alpha^+(s,t) - s(B+Ct)\}$. The decision criterion remains the same in this case.

Using the equivalent-capacity based admission criterion is more convenient. Here, the estimation of the equivalent capacity of the existing flows can be maintained in the background, i.e. the estimate of C_{equ} can be recomputed based on periodic measurements. At the arrival of a new flow, the $C_{\text{equ}}+p \le C$ criterion has to be checked, which differs from (7) only in a correction term that is the peak rate of the new flow. Hence, the timing-sensitive operation (the admission decision) involves only a simple addition and a comparison, while the time-consuming double optimisation can be performed in the background, with more relaxed timing requirements.

The proposed method thus enables the deployment of the many sources asymptotics in practice not only through the reduction of its complexity, but through shifting the computations away from the critical decision instant.

References

[1] C. Courcoubetis, V. A. Siris, and G. D. Stamoulis. Application of the many sources asymptotic and effective bandwidths to traffic engineering. *Telecommunication Systems*, 12:167–191, 1999.

[2] C. Courcoubetis and R. Weber. Buffer over ow asymptotics for a buffer handling many traffic sources. *Journal of Applied Probability*, 33:886–903, 1996.

[3] N. G. Duffield. Economies of scale for long-range dependent traffic in short buffers. *Telecommunication Systems*, 7:267–280, 1997.

[4] R. J. Gibbens and Y. C. Teh. Critical time and space scales for statistical multiplexing in multiservice networks. In *Proceedings of International Teletraffic Congress (ITC)*, pages 87–96, Edinburgh, Scottland, 1999. ITC'16.

[5] F. P. Kelly. Notes on effective bandwidths. *Stochastic Networks: Theory and Applications*, 4:141–168, Oxford University Press, 1996.

[6] J. T. Lewis, R. Russell, F. Toomey, S. Crosby, I. Leslie, and B. McGurk. Statistical properties of a near-optimal measurement-based CAC algorithm. In *Proceedings of IEEE ATM*, pages 103–112, Lisbon, Portugal, June 1997.

[7] M. Montgomery and G. de Veciana. On the relevance of time scales in performance oriented traffic characterizations. In *Proceedings of the Conference on Computer Communications (IEEE INFOCOM)*, volume 2, pages 513–520, San Francisco, USA, March 1996.

[8] I. Norros. A storage model with self-similar input. *Queueing Systems*, 16(3/4):387–396, 1994.

[9] P. Tran-Gia and N. Vicari, editors. *Impacts of New Services on the Architecture and Performance of Broadband Networks - COST257 Final Report.* compuTEAM 2000, 2000.

Appendix: The Improved Buffer Requirement Estimator

Let us introduce $L(s,t) \overset{\text{def}}{=} t\,\alpha(s,t) + \frac{\gamma}{s} - Ct$, resulting from the isolation of the variable B from $J(s,t) = -\gamma$. That is, $L(s,t) = B$ holds after the rearrangement. Similarly to (4) and (11) the critical space and time scales of

$$\widetilde{B}_{\text{req}} \overset{\text{def}}{=} \sup_{t>0} \inf_{s>0} L(s,t) \tag{22}$$

are $s'(t) \overset{\text{def}}{=} \arg\inf_{s>0} L(s,t)$, $t' \overset{\text{def}}{=} \arg\sup_{t>0} K(s'(t),t)$ and $s' \overset{\text{def}}{=} s'(t')$. The extremiser pair of (22) is then (s',t'), like in the previous cases.

Analogously to the equivalent capacity, it is also true that the buffer requirement defined by the triple optimisation in (8) $\widetilde{B}_{\text{req}} = B_{\text{req}}$ holds. Consequently, only two optimisations are needed instead of three to determine the buffer requirement B_{req}, matching the case of the equivalent capacity.

The statements of Sect. 3.1 and Sect. 3.2 can now be reformulated.

Theorem 2. *(12) ⇔ $L(s',t') < B$, (13) ⇔ (14) ⇔ $L(s',t') = B$ and (15) ⇔ (16) ⇔ $L(s',t') > B$.*

Lemma 4. *(17) ⇔ (18) ⇔ $L(s'(t),t) < B$, (19) ⇔ (20) ⇔ $L(s'(t),t) = B$, (21) ⇔ $L(s'(t),t) > B$.*

Corollary 2. *The buffer requirement defined by the double optimisation in (22) equals the one defined by the triple optimisation in (8): $L(s',t') = \widetilde{B}_{\text{req}} = B_{\text{req}}$.*

Proposition 2. *If one of the double optimisations (3), (10) and (22) has a unique extremising pair and $J(s^*,t^*) = -\gamma$ (or $K(s^\dagger,t^\dagger) = C$ or $L(s',t') = B$), then the three extremiser pairs coincide, $t^* = t^\dagger = t'$ and $s^* = s^\dagger = s'$.*

Proof. The proofs follow the structure of those in Sect. 3.1 and Sect. 3.2. □

QoS with an Edge-Based Call Admission Control in IP Networks

Daniel R. Jeske, Behrokh Samadi, Kazem Sohraby, Yung-Terng Wang, and Qinqing Zhang

Bell Labs, Lucent Technologies
Holmdel, New Jersey 07733, USA

Abstract. Central to the viability of providing traditional services over IP networks is the capability to deliver some level of end-to-end Quality of Service (QoS) to the applications and users. IP networks continue to struggle to migrate from a cost effective best effort data service solution to revenue generating solutions for QoS-sensitive applications such as voice and real-time video. For the case of a network of Media Gateways controlled by SoftSwitches, we propose the use of a measurement based call admission control algorithm at the edge of the network as an approach to provide a cost effective QoS solution. The proposed method utilizes statistical prediction techniques based on available performance measurements without complex QoS management of the packet network. Simulation analysis shows that significant gains in QoS can be achieved with such an edge-to-edge measurement based approach.

Keywords: QoS, VoIP, CAC, SoftSwitch

1. Introduction

One of the notable recent advances in converged networks is the development of the SoftSwitch (SS) technology. With SS technology, control plane functions and interworking between packet and circuit switched network signaling services can be implemented with standard based protocols and used to provide among other things, a locus of resource management of the bearer channels through circuit and packet switched networks [1]. Applications of SS technology include the Voice tandem solution for the Public Switched Telephone Network (PSTN) and VoIP solutions for IP end points. A reference network architecture is shown in Fig.1.

For the voice Tandem solution, voice calls originating and terminating within PSTN would be handled by signaling the ingress and the egress SoftSwitch, and then the destination PSTN switch to complete the call setup procedure. While management of voice QoS in the PSTN is well understood, a different matter is providing QoS when a packet network is used as the transport between the gateways, or when IP end points such as SIP or H.323 devices get involved with the call.

Central to the viability of providing traditional services over IP networks is the capability to deliver some level of end-to-end QoS to the applications and users. Many solutions have been proposed and implemented for ATM based packet networks and standards. On the other hand, IP based networks continue to struggle

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 178-189, 2002.

to migrate from cost effective best effort data service solution to revenue generating QoS solutions for more demanding applications such as voice and real-time video.

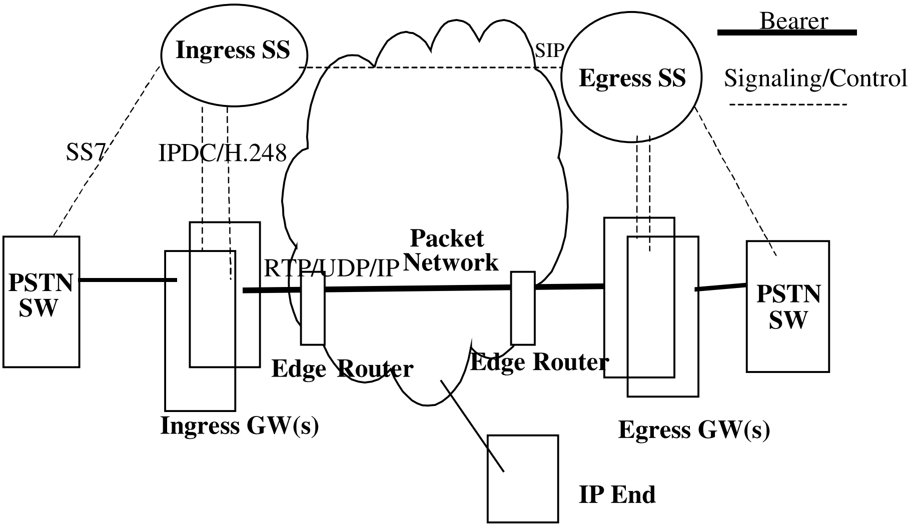


Fig. 1 . Reference Network Architecture

In this paper, we explore the possibility of utilizing available measurements at the edge of packet networks by proposing a distributed edge-to-edge measurement based call admission control (CAC) to provide a cost effective QoS solution. Controlling variables are provided to allow the service providers to influence the tradeoffs of network resource utilization and risks of compromised QoS. The proposed method utilizes statistical prediction techniques using available performance measurements. In Section 2, we discuss the framework of QoS support in IP networks. In Section 3, we present details of the statistical prediction techniques and their application to a measurement based CAC algorithm. Quantitative analysis of the achieved QoS is reported in Section 4 with comparisons to a "best case," where the packet network resources are closely managed to provide a known capacity (see e.g. [2]), and a “worst case,” where every call is admitted.

2. Quality of Service Support

For ease of exposition, we use voice service as the application throughout this paper although many of the principles can be extended to include other applications such as multimedia calls. The basic characterization of QoS for the voice application is well-studied (see e.g., [3]). Performance metrics often used to measure QoS for the bearer or user plane of packet telephony include: end-to-end packet delay, delay jitter and packet loss. Note that there are other performance and QoS metrics such as call setup delay, post-dial delay and ring-back delay which are outside the scope of this paper.

The desired end-to-end delay is usually very small for toll quality, the recommended one-way value being 150 ms [4]. Delay jitter is the variation in the delay of consecutive packets. For streaming applications, the delay jitter should be small enough so that the traffic stream can be delivered at a constant rate to the receiver to prevent packet loss. If the delay jitter increases beyond a limit, the packet is regarded as lost. Usually some play-out buffer mechanism at the receiver is provided to absorb the jitter. Buffering adds to the delay and needs to be dimensioned carefully. Packet loss can also occur as a result of buffer overflows in the network. The degree of QoS degradation due to packet loss depends on the application and the coding schemes.

We now turn our attention to the issue of primary interest of this paper: providing QoS support over the packet network. The most developed paradigm is ATM QoS, which includes several key principles that are applicable beyond ATM networks (see e.g. [5,6]). Networks that are IP based are evolving toward use of DiffServ/MPLS which provides the ability to manage QoS through traffic engineering of MPLS Label Switch Paths (see e.g., [7]) and Differentiated Service markings at the edge. However, before these technologies are ubiquitous and network management solutions get developed and deployed, the service providers are left with no choice but to over-provision their networks in order to minimize the risk of degrading QoS.

An economic alternative, which does not require any network resource availability information, is to utilize the measurements available at the edge of the network in the CAC to make the best prediction of whether or not to admit a new call. The objective is as usual: admit as many calls as possible while supporting adequate QoS of the calls already in progress. An obvious implication of not having proactive control of network resources is that there is a period during which the network is vulnerable to sudden changes in congestion levels. We address this and identify other critical issues associated with using available QoS measures in connection with CAC.

3. Measurement Based CAC

In our proposed CAC scenario, both the ingress and egress SS use historical QoS measurements from the originating and terminating Media Gateways (MGW) to make a QoS prediction for each new call. In this paper, we focus on one-way packet loss rate as the primary QoS measure since it is readily available from most MGWs. If either of the one-way predicted QoS measures is unsatisfactory, the new call is rejected, otherwise it is accepted. To implement this approach, sufficiently accurate predictions for one-way packet loss rates are required.

In this section, we describe two statistical models for predicting packet loss rates. The first is a simple exponentially weighted moving average (EWMA) model. The second is an auto-regressive (AR) model which is more sophisticated and was evaluated for use so that we could determine if the EWMA model was overly simplistic. After providing some motivation for each model, we fit each model to traces that were collected, and conclude that the extra complications associated with the AR model may not be justified.

3.1 EWMA Model

Let the sequence $\{Z_k\}_{k=-\infty}^{T}$ denote one-way packet loss rates associated with calls completed over the same path that a new call would utilize if it were admitted into the network. Alternatively, without additional complication, $\{Z_k\}_{k=-\infty}^{T}$ could represent observed aggregate (over all calls on the path) packet loss rates over consecutive and disjoint windows of a prescribed width. Intuitively, one would expect correlation among the Z_k values, and moreover, a degree of non-stationary behavior. A widely used correlation model that accounts for a slowly varying time-dependent mean is a auto-regressive integrated moving average model (ARIMA) [9] which writes:

$$Z_k - Z_{k-1} = \varepsilon_k - \theta\varepsilon_{k-1} \quad . \tag{3.1}$$

Here, the sequence $\{\varepsilon_k\}$ is a white noise process with zero mean and variance σ^2. $|\theta| < 1$ is a parameter that dictates the correlation amongst the Z_k values. In the case of a slowly varying mean function for $\{z_k\}$, the mean of the left-hand-side of (3.1) is zero and the sequence of first-order differences, $D_k = Z_k - Z_{k-1}$, is a stationary process. It is easy to verify that θ is the lag-1 auto-correlation of the $\{D_k\}$ sequence.

At time *T*, the best predictor of Z_{T+k} ($k \geq 1$) is the conditional [given $\{Z_k\}_{k=-\infty}^{T}$] expected value of Z_{T+k}, say $\hat{Z}_{T+k}$. It can be shown [9] that $\hat{Z}_{T+k} = (1-\theta)Z_T + \theta\hat{Z}_T$, and $\hat{Z}_{T+k}$ is a weighted average of the $\{Z_k\}_{k=-\infty}^{T}$ sequence. Moreover, the weights die off exponentially. Consequently, $\hat{Z}_{T+k}$ is frequently referred to as an exponential weighted moving average (EWMA) of the $\{Z_k\}_{k=-\infty}^{T}$ sequence. Note that under the EWMA model, the *k*-step ahead predictor is independent of *k*. This is a convenient result for our CAC application. In particular, if *T* calls have completed when a new call arrives, the new call will not be the (*T*+1)st completed call in the sequence, but rather will be the (*T*+*D*+1)st completed call in the sequence, where *D* is a random variable representing the number of completed calls during the holding time of the new call. What we need is $\hat{Z}_{T+D+1}$, which by the above property is simply $\hat{Z}_{T+1}$. With the EWMA model, we are thus able to avoid the difficulty associated with *D* not only being unknown, but also being a random variable.

For many EWMA applications, the value of θ is chosen based on intuitive feelings of how much the past should be weighted. If Z_k is rapidly changing, then $\theta = 0.2$ is a common choice, while if they are not changing very fast $\theta = 0.8$ is a common choice. Since θ is the lag-1 auto-correlation of the $\{D_k\}$ sequence, it could be estimated periodically from the observed packet loss measurements, possibly even using a sliding-window scheme. Alternatively, θ can be computed adaptively, changing at each prediction epoch (a new call arrival in our case) [10,11].

3.2 AR Model

An alternative to the EWMA model (3.1) is a p-th order auto-regressive model, AR(p). The AR(p) model relates the current value of a process to the proceeding p values and a white noise innovation term. In particular, we have

$$Z_t - \mu = \sum_{k=1}^{p} \phi_k (Z_{t-k} - \mu) + \varepsilon_t \tag{3.2}$$

where μ is the mean of the $\{Z_k\}$ process and the $\{\phi_k\}_{k=1}^{p}$ are the so-called auto-regressive coefficients. Unlike the EWMA model, the AR(p) model assumes the $\{Z_k\}$ process is stationary. (Recall that the EWMA model assumes the first-order differences of the $\{Z_k\}$ process is stationary.)

At time T, the best-unbiased predictor of an observation D+1 steps into the future, say $\hat{Z}_{T+D+1}$, is the conditional expected value of Z_{T+D+1}, given $\{Z_k\}_{k=-\infty}^{T}$. It can be shown [9] that $\hat{Z}_{T+D+1} = \mu + \sum_{j=1}^{p} \pi_j^{(D+1)} (Z_{T+1-j} - \mu)$, where coefficient $\pi_j^{(D+1)}$ is computed in a bootstrap recursive manner according to: $\pi_j^{(1)} = \phi_j$, $\pi_j^{(2)} = \phi_1 \pi_j^{(1)} + \phi_{j+1}$, $\pi_j^{(3)} = \phi_1 \pi_j^{(2)} + \phi_2 \pi_j^{(1)} + \phi_{j+2}$, … , $\pi_j^{(D+1)} = \sum_{i=1}^{D} \phi_i \pi_j^{(D+1-i)} + \phi_{j+D}$. Unlike the EWMA model, the prediction equation explicitly depends on the unknown random variable D, as is clearly evident by the general form of $\hat{Z}_{T+D+1}$. The best we can do is replace D by its mean value, introducing another source of variability into the prediction error.

Use of $\hat{Z}_{T+D+1}$ requires an estimate of μ and $\{\phi_k\}_{k=1}^{p}$. Moreover, these estimates must be updated periodically to reflect the changing conditions of the underlying network. Let $Z_1, \dots, Z_n$ denote the set of observations corresponding to a particular update interval. The estimate of μ is simply $\bar{Z} = \sum_{t=1}^{n} Z_t / n$ and method of moment estimators of $\{\phi_k\}_{k=1}^{p}$, say $\{\hat{\phi}_k\}_{k=1}^{p}$, can be obtained by solving the $p x p$ linear system of Yule-Walker equations [9] which utilize the first p sample autocorrelations.

In order to estimate a sample autocorrelation reliably from a statistical point of view, at least 50 observations are recommended, implying $n > 50 + p$. On the other hand, the time required to collect n observations is n / λ, where λ is the call arrival rate (arrivals/minute), and we need $n / \lambda < S$, where S is the duration (minutes) over which we are willing to assume the network is stationary. Hence, we require $(50 + p) < n < \lambda S$, an interval constraint on the magnitude of n. (For example, if p=10, $\lambda = 10$ calls/minute and $S = 10$ minutes, then we would require $60 < n < 100$.)

An additional constraint linking the call arrival rate, the call holding time and the duration of the stationarity interval derives from the following conditions. First, the new call arrival should complete within the interval of stationarity, and second, the p calls used for the packet loss prediction for the new call arrival should arrive and complete within the interval of stationarity (see Figure 2). It follows that $p / \lambda + 2HT < S$, where HT is the average call holding time. (For example, if p=10, $\lambda = 10$ calls/minute, and $S = 10$ minutes, then we must have $HT < 4.5$ minutes.)

Since AR(*p*) has a higher dimensional parameter space than EWMA, we could expect more precise predictions from it. However, its application is hard in that *D* is a random variable and the best we can do is replace it by its expected value, $\lambda \times HT$, which would seem to offset some of the additional precision. Moreover, it is clear that the AR(*p*) model is more cumbersome to implement. In particular, the need to frequently update the parameter estimates is a drawback since at each update epoch, the Yule-Walker equations must be solved and then the bootstrap recursive scheme must be used to obtain the necessary prediction coefficients required by $\hat{Z}_{T+D+1}$. These observations raise the question as to whether or not the extra precision seemingly available from the AR(*p*) model is significant enough to justify its added complexity. We examine this question in the next section by fitting each model to two different packet loss traces and comparing their respective prediction capabilities.

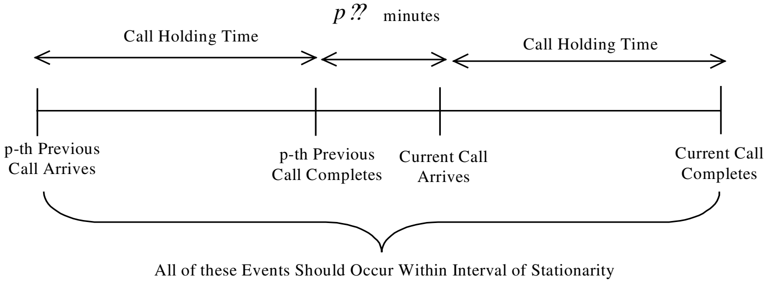

p ?? minutes
Call Holding Time
Call Holding Time
p-th Previous Call Arrives
p-th Previous Call Completes
Current Call Arrives
Current Call Completes
All of these Events Should Occur Within Interval of Stationarity

Fig. 2. Feasibility Condition for AR(*p*) Model

3.3 Empirical Comparison of Models

To compare the relative prediction accuracy of the EWMA and AR(*p*) models, we applied them to two packet loss traces which were collected from two different IP networks. Below we describe the traces and the results obtained from the fitting analyses. We show that the simpler EWMA models compete very well with the AR(*p*) model, thereby making a case for their use in real applications.

3.3.1 University of Massachusetts Trace. The first packet loss trace we examined was obtained from the University of Massachusetts at Amherst (UMASS). The trace is one of many traces collected by sending out packet probes (RTP headers) along unicast and multicast end-to-end connections over the public Internet [8]. The packets were sent out every 20*ms* between UMASS and UCLA. The trace is a binary time series with zeros and ones indicating whether the packet probe arrived successfully or not. There are total 358,000 packets in the trace (a two-hour trace). Post processing on the trace divided it into 179 time intervals. Each time interval represents about 40 seconds and 2000 packets. The packet loss percentage of each interval was used as Z_k, in our analysis. Since there are no overlapping intervals, the analysis is window-based in terms of the generation of the time series and the prediction. The estimation algorithms, EWMA or AR(*p*), apply to either case.

Figure 3 shows the analytical results of the packet loss estimation using three different models. AR(5) is a 5th order AR prediction model with coefficients (0.65, -0.26, 0.34, -0.07, 0.22). SEWMA is a static EWMA with fixed weight $\theta = 0.75$. AEWMA is an adaptive EWMA with an average weight $\theta = 0.75$. The circles are the actual packet loss for each interval. The results demonstrate strong predictive power in this data trace. The 5th order AR model implies that the correlation goes back at least 10,000 packet, or 200 seconds. The AR model predicts slightly better than the two EWMA models. But simpler EWMA model competes well.

Figure 4 shows the prediction errors from the AR(5) model are unbiased. Although there is a fair amount of variance in the prediction error, the algorithm predicts well enough to indicate when the packet loss is likely to be "high".

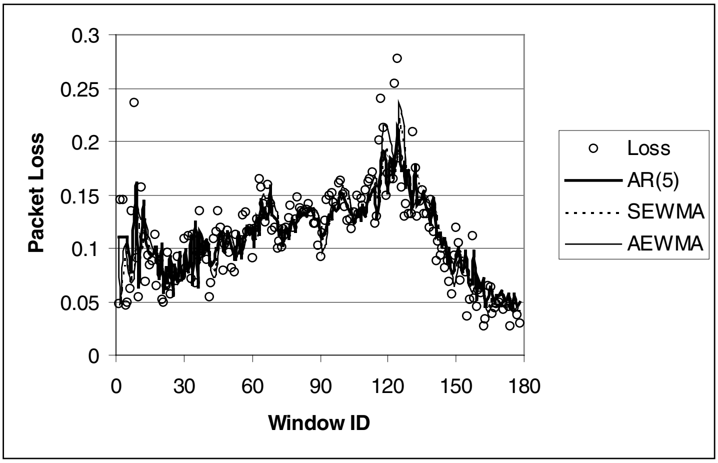


Fig. 3. Analysis of a UMASS trace using different models

3.3.2 NetMeeting Audio Trace. To further evaluate the prediction models, we conducted another experiment to generate a voice over IP trace. The experiment was set up between two desktop computers, located in New Jersey and California. A presentation was sent through NetMeeting over Lucent's Intranet. The talk was encoded by PCM (Pulse Coded Modulation) with a sampling interval of 30 *ms* into a 64 *kbps* audio stream. A commercial software tool, NetXray, was used to capture the packets transported over two end points. Another software tool, Xdecode, was used to process and analyze the captured packets trace. Xdecode reads Ethernet capture files and decodes them in an ASCII format. We further processed the decoded file using shell scripts to identify and subtract the RTP/UDP packets that carried the audio stream. We obtained the RTP packet's sequence number and relative delay offset and generated the corresponding binary packet loss trace. The trace we analyzed had 50,000 packets and lasted about 25 minutes. We divided the trace into 100 intervals of 500 packets each. The corresponding time series $\{Z_k\}$, was thus generated with the packet loss rate of each interval as the dependent variable. Again, the estimation model is window based instead of per call based.

Figure 5 shows the analytical results of the predication using three models. AR(2) is a 2nd order AR predication model with coefficients (1.12, -0.17), SEWMA is a static EWMA with weight $\theta = 0.24$. AEWMA is an adaptive EWMA with average weight $\theta = 0.24$. The actual packet rates, denoted by the circles, show jumps between low packet loss and high packet loss. The results demonstrate strong prediction power in this data set also. This time the correlation goes back at least 1000 packets, or 30 seconds. The AR(2) model performs a little better than the EWMA models, particularly for the very clean transitions from low high loss. The EWMA models compete fairly well.

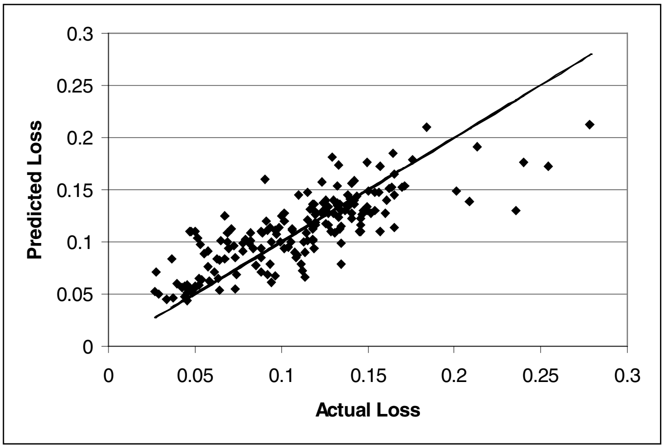


Fig. 4. Prediction error from the AR model on a UMASS trace

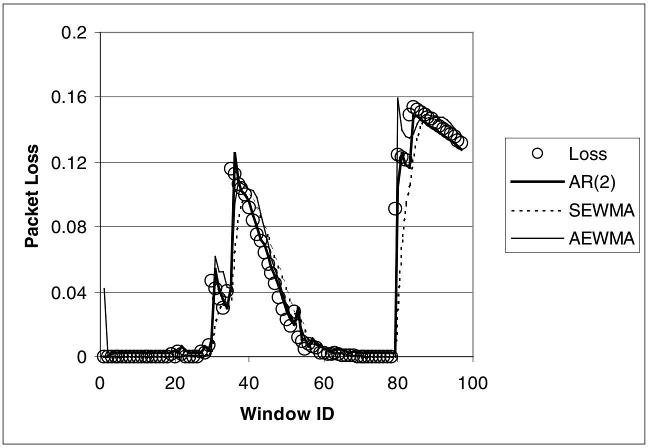


Fig. 5. Analysis of a NetMeeting trace using different models

Figure 6 shows the prediction error from the AR(2) model. The variance appears smaller than the UMASS trace. It is very likely that the predication error variance is proportional to the overall congestion level.

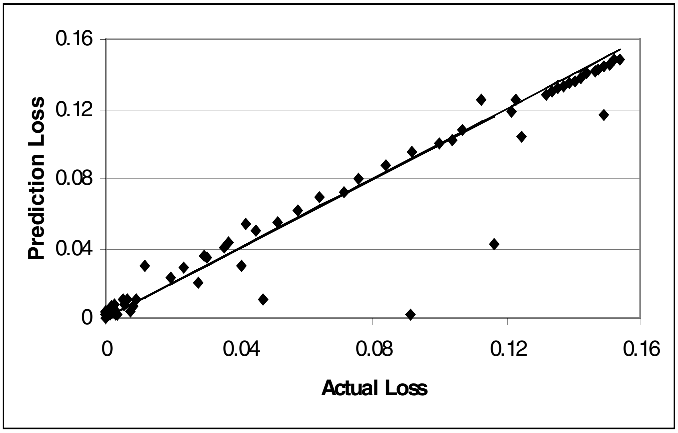


Fig. 6. Prediction error from the AR model on a NetMeeting trace

3.4 CAC Options

We now describe two variations of our proposed CAC approach, which differ in how the historical packet loss observations are calculated. The difference is significant in that the second option requires more information out of the MGWs. In Section 4 we compare the effectiveness of the two alternatives and evaluate whether or not the additional information significantly improves the overall QoS.

CAC-1: In this option, packet loss rate is defined as the percentage of packets that are lost due to network conditions such as buffer overflow, and lost or corrupt connections. The process of determining the packet loss rate for the completed calls is through EWMA using a fixed smoothing factor of 0.8. If at the time of the new call arrival, neither of the predicted one-way packet loss rates exceeds a given threshold (say, 1%), the new call is accepted, otherwise it is rejected. For the calls that are accepted, overall QoS is measured as the ratio of the completed calls for which the actual packet loss *or* jitter delay is less than a given threshold (say 2%). In order to discount the effect of packet loss in the admission process when there are long inter-arrival times between consecutive calls, the packet loss rate is discounted at a rate of say 50% if the update takes place in longer than 10 seconds. Also, in order to account for calls with very long call holding times (longer than say 100 seconds), the packet loss is determined only over the last 100 seconds of the call.

CAC-2: In this option, packet loss rate is defined as the percentage of lost packets due to all of the reasons defined for CAC-1 plus jitter variation. There is clearly an additional cost to acquire jitter measurement. However it is recognized as an important component of overall packet loss at the application layer and many vendors do support it. Note also that if network buffers are large, delay jitter may be the dominant component of packet loss. Our intent with CAC-2 is to ascertain that the use of delay jitter in the calculation of packet loss significantly improves the effectiveness of the CAC algorithm. All other details associated with CAC-2 are identical to CAC-1.

4. Performance of CAC Algorithms

A simulation model is used to compare CAC-1, CAC-2 with two other scenarios. First, a "no CAC" scenario, in which *all* calls are admitted into the network. Second, a "Static CAC" scenario, where the maximum number of calls that can be allowed on a path is predetermined from provisioned bandwidth, and the number of calls in-progress is consulted before a new call is admitted.

In the simulation model calls arrive to a single server queue, representing the network with no other interfering traffic. Each call generates voice packets every 20 ms from a PCM encoded source at 64 Kbps. Call inter-arrival times are exponentially distributed with a rate of 0.42 calls per second. The service rate of the queue corresponds to a transmission link with a capacity of 5 Mbps. The service time of a packet is therefore 1280/5000 = 0.256 ms. The call holding time is also assumed to be exponentially distributed with a mean of 180 seconds. The queue capacity is varied in the simulations. The three QoS measures are: 1) good call rate, where a good call is defined as those for which the packet loss rate over the entire duration of the call does not exceed a threshold (2% is used in the experiments), 2) call blocking rate, defined as the percentage of new call arrivals that were denied admission and, 3) packet loss, measured as the total number of packets lost due to buffer or delay jitter, and the average packet loss is determined as the percentage of packets lost within the duration of a call.

4.1 Simulation Results

Table 1 shows call blocking rate (B), good call rate (Q), and average packet loss (L) for buffer sizes of 200, 400, 600 and infinite (packets). The two entries associated with L represent the packet loss components due to buffer overflow and delay jitter, respectively. Note that since CAC-1 relies only on packet loss due to buffer overflow, the performance for the infinite buffer size case is identical to the "No CAC" case. For the simulations summarized by Table 1, play-out delay was 60 ms. In this scenario the results for Static CAC, independent of the buffer size, are B=6.2%, Q=100% and L=0%. In the case of "No CAC," all calls are allowed to the system, so obviously B=0. As a result, the quality of service as measured by Q and L are the worst compared to the other options. CAC-1 and CAC-2 both improve Q, particularly when the buffer size is small. Note that in that case, network buffer overflow dominates the overall packet loss rate so the difference between CAC-1 and CAC-2 is minimal. For large network buffers, delay jitter contributes to the overall packet loss, and CAC-2 results in better Q values compared to CAC-1, although the gains are modest for these cases. Of course CAC-2 always has higher B values since its predictions for packet loss will always be larger than those of CAC-1.

Table 2 shows the simulation results for a 20 ms play-out delay and for buffer sizes of 200 and infinity. Comparing Table 1 and Table 2 for a buffer size of 200 packets, we note that the smaller play out delay translates into difference in the performance of CAC-1 and CAC-2. In Table 2, delay jitter is a significant component of the overall packet loss rate, and thus the CAC-2 packet loss predictions will have better fidelity. As a result, both B and Q are higher for CAC-2, and L is appreciably lower. Although Q falls off precipitously from Table 1 to Table 2 for both CAC-1 and CAC-

2, Table 2 still shows that both options improve upon the "No CAC" case. Similar to the previous case, the results for Static CAC, independent of the buffer size, are B=6.2%, Q=100% and L=0%.

Table 1. Performance of CAC options (60 ms play out delay)

Scenario	Buffer Size (Packets)			
	200	400	600	Infinity
No CAC	B: 0% Q: 86.6% L: 2.2%, ~0%	B: 0% Q: 49.9% L: 2.1%, 17.5%	B: 0% Q: 49.9% L: 2.1%, 19.1%	B: 0% Q: 41.4% L: 0%, 34.1%
CAC-1	B: 5.9% Q: 95.1% L: ~0%, ~0%	B: 6.0% Q: 55.9% L: 0.5%, 10.7%	B: 5.9% Q: 53.8% L: 0.5%, 13.2%	B: 0% Q: 41.4% L: 0%, 34.1%
CAC-2	B: 5.9% Q: 95.1% L: ~0%, ~0%	B: 11.4% Q: 61.7% L: 0.2%, 6.9%	B: 12.1% Q: 62.9% L: 0.2%, 8%	B: 12.5% Q: 61.8% L: 0%, 9.8%

Table 2. Performance of CAC options (20 ms play out delay)

Scenario	Buffer Size (Packets)	
	200	Infinite
No CAC	B: 0% Q: 48.7% L: 2.2%, 19%	B: 0% Q: 43.6% L: 1.7%, 31.5%
CAC-1	B: 5.9% Q: 52% L: 0.5%, 12.5%	B: 0% Q: 43.6% L: 1.7%, 31.5%
CAC-2	B: 11.8% Q: 59.7% L: 0.2%, 6.7%	B: 13.3% Q: 60.6% L: 0%, 10.3%

In general, as expected, simulations indicate that both CAC-1 and CAC-2 increase Q compared to the "No CAC" case. What is worth noting however is that the performance of both CAC-1 and CAC-2 are approaching that of the best case "static CAC" when the network buffer and play-out buffers are sized appropriately. We have also noted that the difference between CAC-1 and CAC-2 with respect to Q is modest. However, with respect to L, CAC-2 provides appreciable reductions compared to CAC-1 and significant reductions compared to "No CAC".

5. Summary

In this paper, we presented a distributed edge-to-edge measurement-based approach to QoS support for a VoIP application. The measurements, namely packet loss rates, are available on most media gateways. We compared two statistical prediction methods: AR and EWMA, to predict packet loss from these measurements. It was demonstrated that the less complicated EWMA approach produced similar results under various traffic conditions on a public IP network.

Two measurement based CAC methods were analyzed and compared through simulations with encouraging initial results. Both methods use predicted packet loss, with the difference that the packet loss is measured on different interfaces. CAC-1 uses measurements collected on the network layer and thus only represents packet loss due to various network conditions. CAC-2 uses the measurements collected on the application layer at the time where the packet play out is to take place. We quantified the magnitude of improvements with CAC2.

These observations help us determine the details of algorithms and identify the required interfaces to the network elements. Future work includes: (i) extension of the simulation study to include data sources and multiple MGWs, (ii) investigation of the nature of packet loss in typical network environments to better differentiate CAC-1 and CAC-2, (iii) development of a revenue model based on QoS measures to facilitate comparison of alternative CAC scenarios, and (iv) investigation of the use of other measures, such as delay, in CAC algorithms.

References

1. R.A. Lakshmi, “ The Lucent Technologies Softswitch: Realizing the promise of convergence,” Bell Labs Technical Journal V4, 2, APR-JUN, 1999, p 174-195
2. B. Doshi, E. Hernandez-Valencia, K. Sriram, YT Wang and O.C. Yue, "Protocols, Performance, and Controls for Voice over Wide Area Packet Networks," Bell Labs Technical Journal, Vol 3, No. 4, Oct 98.
3. K. Sriram and Y. T. Wang, "Voice over ATM using AAL2 and Bit Dropping: Performance and Call Admission Control," IEEE Journal of Selected Areas in Communications, Vol.17, No.1, 1999, pp.18-28
4. ITU-T Recommendation G.114, One-Way Transmission Time, 2/96
5. M. Grossglauser and D. Tse, "A Framework for Robust Measurement Based Admission Control,” IEEE/ACM Transactions on Networking V7, 3, JUN, 1999, p293-309
6. Z. Dziong, M. Ji, and Y.T. Wang, "Learning Algorithm for CAC adjustment in ATM networks", ATM/IP workshop, June 1999
7. P. Aukia, M. Kodialam, P. Koppol, T. Lakshman, H. Sarin, and B. Suter. RATES: A server for MPLS traffic engineering,” IEEE Network Magazine, pp. 34--41, March/April 2000
8. M. Yajnik, S. Moon, J. Kurose, and D. Towsley, “Measurement and Modelling of the Temporal Dependence in Packet Loss,” Proceedings of Inforcom99.
9. W.S. Wei, Time Series Analysis, Addison-Wesley Publishing Company, 1990.
10. D.W. Trigg and A.G. Leach, “Exponential smoothing with adaptive response rate,” Operations Research Quarterly, Vol.18, Issue 1, 1967, pp.53-59
11. D.Jeske, W.Matragi and B.Samadi, “Adaptive Play-out Algorithms for Voice Packets in a LAN Environment,” International Conference on Communications (ICC), 2001.

Admission Control and Capacity Management for Advance Reservations with Uncertain Service Duration

Yeali S. Sun[1], Yung-Cheng Tu[2], and Meng Chang Chen[3]

[1] Dept. of Information Management
National Taiwan University
Taipei, Taiwan

[2] Dept. of Computer Science
National Tsing Hua University
Hsinchu, Taiwan

[3] Institute of Information Science
Academia Sinica
Taipei, Taiwan

Abstract. Different from Immediate Request (IR) service in packet-switched networks, admission control for Advance Reservation (AR) service is more complex - the decision points include not only the start time of the new connection, but also the instants that the new connection overlaps with connections already admitted in the system. Traditional approach on advance reservation considers only a fixed scheduled period. When overtime occurs (often quite approaching the end of the originally scheduled service period) depending on network load and resource usage, the service may easily be disrupted due to insufficient resources available. Examples include the broadcasting of sports events and business video-conference calls. In this paper, we study the problem of admission control and resource management for AR service with uncertain service duration. The objective is to maximize user satisfaction in terms of service continuity and guarantee of QoS while minimizing reservation cost and call blocking probability of the AR service. An innovative two-leg admission control and bandwidth management scheme is proposed. Service continuity, user utility and reservation cost functions are proposed here to evaluate user's satisfaction and the efficiency of resource allocation. Simulation results are presented.

1 Introduction

Many signaling and admission control designs of the quality of service (QoS) support in packet-switched networks such as RSVP [1] focus on requests that must be served immediately, commonly known as Immediate Request (IR) service. In today's Internet, there is demand for Advance Reservation (AR) service. For example, many important business conference meetings and calls are pre-planned and scheduled. By advance reservation service, users can know whether they can

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 190–201, 2002.

get full QoS support of their communication needs over the Internet in advance. From the service provider's perspective, knowing the future needs ahead allows them to better manage the allocation and sharing of network resources between users, and to serve their customers in a more affirmative, predictable way.

In order to perform admission control and resource allocation, requests for advance reservation must specify three basic data: service start time, QoS requirement and duration of service. Recently a few works were proposed. They all assume these parameters are given and of fixed value when requests are submitted [2,3,4,5,6,7,8]. In reality, these information may not be known in prior, especially the service duration. Examples include the broadcasting of sports events and business videoconference calls. Typically, there is so-called scheduled duration, e.g., two hours for a broadcast sports event. But often there are overtimes. Traditional approach on advance reservation considers only a fixed scheduled period. When overtime occurs (often quite approaching the end of the originally scheduled service period) depending on network load and resource usage, the service may easily be disrupted due to insufficient resources available. Therefore, it becomes a challenge to the service provider to fulfill the needs of such types of requests assuring both the continuity of service and guarantee of QoS given the uncertain service duration at the time the request was scheduled while in line with its goal of maximum network resource utilization.

In this paper, we focus on AR request with longer lifetime such as Internet broadcast events and videoconferences. Here, we propose an innovative two-leg admission control and resource reservation scheme for AR requests with uncertain service duration over the Internet. The idea is to perform bandwidth reservation in *multiple* stages. Each stage has a fixed duration and specific level of quality of service to assure. Thus, service provider can efficiently manage network resources and allocate bandwidth necessary to guarantee service quality requirements of individual connections in each stage.

To further tackle uncertainty and to maximize network resource utilization, an update mechanism is used. A convex user utility function is defined to characterize the level of user satisfaction for those admitted AR connections with the combined bandwidth allocation and service continuity. A reservation cost is also defined to evaluate the efficiency of the overall network resource allocation in advance reservation service.

Other works related to advance reservations in the past include extensions to the existing protocols and signaling capabilities, e.g., extension of ST2 protocol [2,9] and RSVP [3]. In [5], the authors proposed a distributed reservation scheme and its possible implementation. In [10], the authors studied AR requests with uncertain duration. It does not address the service continuity problem. In [8], a measurement-based approach is proposed to estimate the bandwidth used for existing connections with fixed duration. In [6,7], they discussed the admission control for connections in progress that are preemptable or interruptible. Specifically, in [6], they studied the issue of resource sharing between AR and IR services. In [7], they gave a general description of the policy and pricing schemes for advance reservations. Most of these works assumed that service durations are fixed and available at the admission control time. In [6,11], they assumed the service times follow some distribution. An estimate or a safe upper bound of the service duration must be given at the request submission time. In this paper, we focus on the admission control and bandwidth allocation problem for AR requests with uncertain service duration.

The organization of this paper is as follows. In Section 2, we present the proposed Two-leg bandwidth reservation scheme. The definitions of service continuity and user utility are given. An update mechanism is also presented. In Section 3, admission control of the proposed scheme is described in detail. In Section 4, reservation cost is presented. In Section 5, simulation results are presented to show the benefits of the proposed scheme. Finally, we give a conclusion in Section 6.

2 The Two-Leg Resource Reservation Scheme

For advance reservation service, there are more than one decision points to check. They include all the time instants the duration of the new connection overlaps with the start time of any connections already admitted in the system. Figure 1 depicts the admission control decision points for AR connections with a fixed duration; the number of decision points is finite. This is, however, not true for the case of uncertain duration.

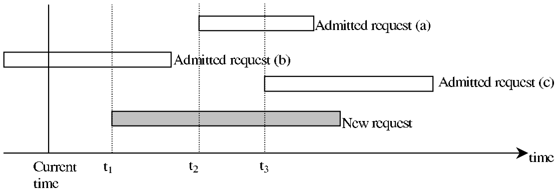


Fig. 1. There may have more than one decision points $\{t_1, t_2, t_3\}$ to consider in the admission control of a new advance reservation request.

In this section, we propose a new admission control scheme with two-leg bandwidth reservation to address the problem of uncertain service duration in AR service. To deal with uncertainty, estimation is used here which is based on the observation that the probability many Internet AR applications will last longer than a duration t is very small when t is sufficiently large (e.g., VCD films information from Blockbuster Homepage [12]). First, we assume for AR requests without specifying service duration, the actual lifetimes will follow certain distributions. Thus, requests can be classified into different categories; each has its own *characteristic lifetime distribution function.* In reality such functions can be obtained through proper data collection, sampling, analysis and characterization from the real world [6].

2.1 Age Function

Let $a_i(t)$ be the probability density function of the nominal duration of type i AR connections and s_i is the start time of the connection. We define the age function of connection i, $A_i(t)$, as the probability that connection will end after time $s_i + t$,

$$A_i(t) = Pr\{duration \geq t\} = \int_i^{\infty} a_i(x)dx, \tag{1}$$

2.2 User Utility

We characterize the level of user satisfaction for those admitted AR connections with the combined bandwidth allocation and service continuity. First, a convex function $s_i(T_i)$ is defined to describe the level of satisfaction in terms of service continuity for connection i which lifetime is T_i and D_i is the nominal duration of the corresponding event, i.e.,

$$s_i(T_i) = \begin{cases} e^{k(\frac{T_i}{D_i}-1)} & \text{if } T_i \leq Di \\ 1 & \text{if } T_i > Di \end{cases} \tag{2}$$

The constant k is used to reflect the weight of such effect. Figure 2 shows the values of (2) under different k's. The larger the k the more utility gain is stressed on the continuity of service especially towards the end of the event. For example, if a live broadcast of a basketball game was initially scheduled for three hours but due to overtimes, the event is in fact three and half hours. The nominal duration is three and half hours. The lifetime of the connection however depends on whether a service extension request is issued, say two hour and 45 minutes after the event. If accepted, T_i is equal to nominal duration; otherwise it is three hours.

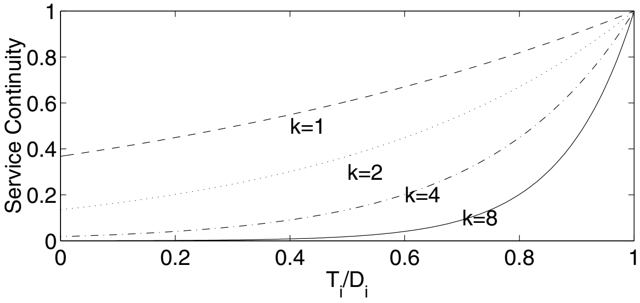


Fig. 2. The values of service continuity under different k's.

Now, we define the *user utility* for connection i as the combination of both service continuity and bandwidth allocation. It is an increasing convex function:

$$u_i(T_i) = \int_0^{T_i} \left(\frac{ds_i(t)}{dt} \times \frac{r_i(t)}{R_i} \right) dt \tag{3}$$

Note that $r_i(t)$ is the bandwidth allocation to connection i at time t and R_i is the requested bandwidth. This function contrasts bandwidth allocated vs. requested during its lifetime continuity of service. The *user utility* is the integral of satisfaction over the nominal service duration. Essentially, the utility value increases when service continues.

2.3 Two-Leg Bandwidth Allocation

Instead of reserving bandwidth indefinitely for connections as in the traditional way, we propose to perform a two-leg admission control and bandwidth reservation for an AR request. The scheme works as follows. Initially, when the first time an AR request is issued *Leg-One admission control* is performed. In this phase, admission control only considers resource allocation for an initial fixed period of time called the *full warranty period* during which the requested bandwidth is reserved for its use if admitted. To handle situations where events may last longer than the warranty periods, a second leg - *Leg-Two admission control* is performed in which an at least minimum amount of bandwidth is reserved at the same time for another fixed period of time called the *at least minimum warranty period.* Admitted AR connection may issue warranty period extension requests at any time afterwards. Additional admission control will be required.

We choose two-leg absolute service warranties than statistical guarantee as in [11]. We believe that this model of advance reservation service is more meaningful because users clearly know the requested QoS is assured during the period. There are several advantages of this model. First, it is easy to implement by service providers. Second, the model is simple enough for the average user to understand so that the users feel comfortable. Known expectations of service assurance reduce risks. Moreover, the administrative cost of tracking usage is low.

Full Service Warranty Period

In the full service warranty period, a connection is assured with full bandwidth allocation. The choice of a good warranty period is essential to the assurance of service continuity. It indeed depends on the nature of the application, i.e., the age distribution function. If a larger value is chosen, the system must reserve resources for a longer period of time. Although this can increase service provider's confidence on service quality delivery and minimize the likelihood of service violation, a major concern is that network resources may be underutilized. Adversely, if a smaller value is used the system can achieve better resource utilization and blocking probability performance. The tradeoff is that more frequent service disruption and lower user satisfaction.

At Least Minimum Bandwidth Reservation for After-Warranty Period

The full service warranty period only represents expected or average duration. The rational behind the design of after-warranty period is to avoid sudden service disruption for connection whose event time is longer than this period, e.g., overtime of sports broadcast events. With resource reservation for the at least minimum warranty period, if a service extension request is rejected, the connection at least has a minimum bandwidth available to continue the service although the quality may degrade. The second leg warranty period is denoted as $D_{i,amw}$. Let parameters $\beta_{i,fw}$ and $\beta_{i,amw}$ be the probability thresholds of the full warranty period $D_{i,fw}$ and at least minimum warranty period $D_{i,amw}$(see Fig. 3).

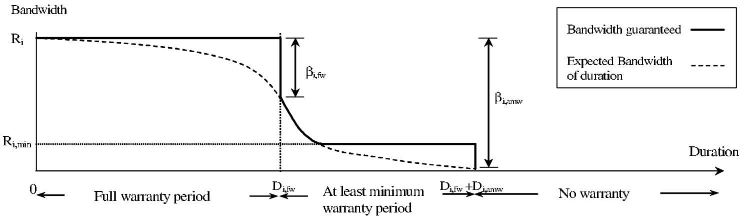


Fig. 3. The amount of bandwidth reservation at different warranty periods.

Compared to full bandwidth reservation, the tradeoff is link utilization. In fact, many Internet applications such as real-time audio/video streaming media are capable of adapting themselves to the network state and can tolerate certain degree of performance degradation. Hence, the bandwidth requirement of an AR request in the proposed service model is given in the form of $< R_i, R_{i,min} >$ where R_i is the bandwidth requirement of the service warranty period and $R_{i,min}$ is the minimum amount of bandwidth acceptable to the connection. The actual bandwidth reservation in the at least minimum warranty period for connection i would be in the range of $< R_i, R_{i,min} >$ (see Fig. 3). R_i can be the effective bandwidth [13,14] or the peak rate.

2.4 Revising Uncertainty with New Data

Uncertain resource allocation is complicated because of the form of "probabilistic" in the duration of which the requested resources are needed as opposed to the fixed duration. Our work focuses on using new data to revise imperfect user-supplied initial knowledge of how long the connection will last. During the course of service, the service provider could periodically poll service user to update his/her knowledge of the connection lifetime or the service user can issue a status update to the service provider notifying whether an extension or early termination of the connection is needed.

3 Admission Control of Full Warranty Period and at Least Minimum Warranty Period

Let the total link capacity designated to the AR service denoted as C_{AR} ($C_{AR} < C_{link}$, C_{link} is the link capacity). Let $A'_i(t)$ is defined for each connection i:

$$A'_i(t) = \begin{cases} 0 & , t < 0 \\ 1 & , 0 \leq t \leq D_{i,fw} \text{(full warranty period)} \\ A_i(t) & , D_{i,fw} < t \leq D_{i,fw} + D_{i,amw} \text{(at least minimum warranty period)} \end{cases}$$

Consider the admission control of a new AR request. Let $< R_{new}, R_{new,min} >$ be the bandwidth requirements of the new connection; $D_{new,fw}$ and $D_{new,awm}$ are the full warranty period and at least minimum warranty service period of the new connection, respectively.

3.1 Leg-One Admission Control

Let P_w be the set of admission control decision points identified, i.e. $P_w = \{t_k, t_k - s_{new} \leq D_{new,fw}\}$, $W(t_k)$ is the set of connections that overlap with new connection at time t_k. The admission decision is based on the following equation:

$$\forall j, j \in W(t_k) \quad \sum_j max(R_{j,min}, (R_j \times A'_j(t_k - s_j)) + R_{new} \leq C_{AR} \tag{4}$$

3.2 Leg-Two Admission Control

Let P_{amw} be the set of admission control decision points identified, i.e. $P_{amw} = \{t_k, D_{new,fw} < t_k - s_{new} \leq D_{new,fw} + D_{new,amw}\}$. $W(t_k)$ is the set of connections that overlap with new connection at time t_k. The admission decision is based on the following equation:

$$\begin{aligned} \forall j, j \in W(t_k) \quad \textstyle\sum_j & max(R_{j,min}, (R_j \times A'_j(t_k - s_j))+ \\ & max(R_{new,min}, (R_{new} \times A'_{new}(t_k - s_{new})) \leq C_{AR} \end{aligned} \tag{5}$$

4 The Reservation Cost

We distinguish two costs for each admitted advance reservation request i: the reservation cost $c_{i,res}$ and actual cost $c_{i,act}$ defined as follows:

$$c_{i,res} = \int_0^{D_{i,amw}} max(R_i \times A'_i(t - s_i), R_{i,min})dt \tag{6}$$

$$c_{i,act} = \int_0^{T_i} max(R_i \times A'_i(t - s_i), R_{i,min})dt \tag{7}$$

Equation(6) is the integral of the total bandwidth reserved to connection i. This is the cost paid by the service provider. Equation(7) is the integral of the bandwidth actually used by connection i. The normalized reservation cost of the system for an interval τ is defined as follows:

$$c_{sys} = \frac{\sum_{i \in AR(\tau)} c_{i,res}}{\sum_{i \in AR(\tau)} c_{i,act}} \tag{8}$$

Its value is no smaller than 1. It will be used to evaluate the performance of the proposed scheme in the next section.

5 Performance Evaluation

In this section, we study the performance of the proposed two-leg advance bandwidth reservation scheme via simulation. The network configuration is shown in Fig. 4. The simulation period is 30 days (we take the daily average statistics from 30 days). For all sets of experiments, the requests are assumed of the type of videoconferences whose nominal service duration is a Pareto distribution with mean 120 minutes and shape=1.8. All requests have the same age distribution function and bandwidth requirements and $< R, R_{min} >=< 1.5Mbps, 256kbps >$

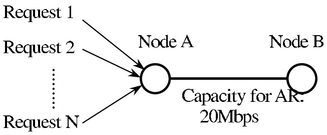


Fig. 4. Network Configuration of the simulation

The arrival process of advance reservation calls is assumed to be a Poisson process. Each call makes a connection reservation with start time in the next day. In each day, we divide 24 hours into "peak zones" (9am-12noon and 2-5pm) and "off-peak zones" (the other times of the day). The probabilities of reservations starting at peak zones or off-peak zones are assumed to be .7 and .3, respectively. Here, we assume the start time of an AR call must be at full or half o'clock (e.g., 9am, 9:30am, etc.). The start time distributions for calls in peak and off-peak zones are all uniform distribution.

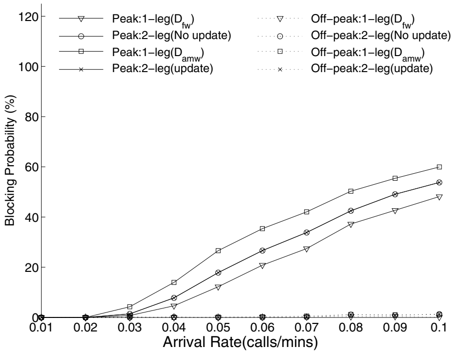
5.1 Service Continuity, User Utility, and Reservation Cost

In this set of experiments, we compare user utility and reservation cost of the proposed Two-Leg bandwidth reservation with that of the traditional one-time reservation approach referred to as one-leg reservation. D_{fw} and D_{amw} are set to 80 minutes ($\beta_{fw} = 0.5$) and 116 minutes ($\beta_{amw} = 0.75$), respectively. Both these two interval values are used as the durations of the one-time reservation for the sake of comparison. In the Two-Leg bandwidth allocation scheme, service update is issued 60 minutes after connection starts. The arrival rate of the AR calls is 0.06 calls/minute. Table 1 shows the user utility. We can see that for connections whose nominal durations are greater than the full warranty period but less than the at least minimum warranty period, in terms of service continuity, it is one under the Two-Leg reservation with or without update. With update, the user utility is further improved. For those connections whose nominal durations are greater than the at least minimum warranty period, the Two-Leg reservation scheme outperforms one-time reservation scheme with D_{fw}. It is as expected that the Two-Leg reservation scheme is not as good as one-time reservation with duration D_{amw} because in the former, the bandwidth allocated after the full warranty period is a function of the bandwidth available at the time service extension request was issued. We know service extension requests often come as short notice and approach the end of the event. How to increase the acceptance probability of the service extension requests is one of the issues that we are currently looking into. In the aspect of reservation cost, the Two-Leg bandwidth allocation scheme performs very well, close to that of one-time with D_{fw}. This implies that the bandwidth reserved in the at least minimum warranty period is efficiently used.

Table 1. Comparisons of service continuity, user utility and reservation cost of the Two-Leg and traditional one-time bandwidth reservation schemes.

	$D_i \leq D_{fw}$		$D_{fw} < D_i \leq D_{amw}$		$D_{amw} < D_i$		
	$s(T_i)$	$u(T_i)$	$s(T_i)$	$u(T_i)$	$s(T_i)$	$u(T_i)$	c_{sys}(24-hours)
1-leg(D_{fw})	1.00	1.00	0.56	0.56	0.12	0.12	1.11
2-leg(No update)	1.00	1.00	1.00	0.64	0.35	0.17	1.16
2-leg(Update)	1.00	1.00	1.00	0.84	0.46	0.22	1.14
1-leg(D_{amw})	1.00	1.00	1.00	1.00	0.35	0.35	1.38

Figure 5 shows comparisons of call blocking probability under different arrival rates for the two schemes. As expected, because the extra bandwidth reservation for at least minimum warranty period in the Two-Leg reservation scheme with or without update, the call blocking probabilities are higher than that of one-time reservation with duration D_{fw} but lower than that of the one-time reservation with D_{amw}. Figure 6 shows the reservation cost. One can see that the reservation costs for the Two-Leg reservation scheme with or without update in peak zones, are close to those in the off-peak zones. Adversely, the reservation costs for the one-leg reservations are much higher than those of two-leg approach. Moreover, in the Two-Leg reservation scheme, the reservation cost when with update is lower that when without update. This is because that the bandwidth reserved after update is much likely utilized, thus lowering the reservation cost.



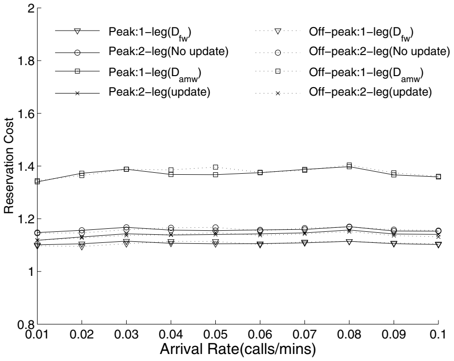
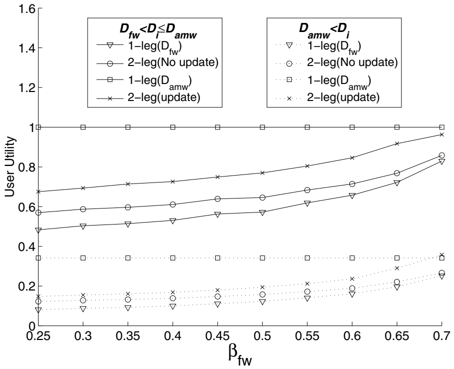


Fig. 5. Comparisons of call blocking probability.

Fig. 6. Comparisons of reservation cost.

5.2 Service Continuity, User Utility, and Reservation Cost

The parameters β_{fw} (D_{fw}) plays an important role in the proposed Two-Leg bandwidth reservation scheme. In this set of experiment, we study the effect of different choices of the full warranty period on user utility and reservation cost. Here, the β_{amw} is fixed and set to 0.75. In Fig. 7, even with update in the Two-Leg reservation scheme, the improvement is limited. This is again because if the update is issued late during the connection lifetime, the blocking probability is likely high. Figure 8 shows the reservation cost under different full warranty periods.



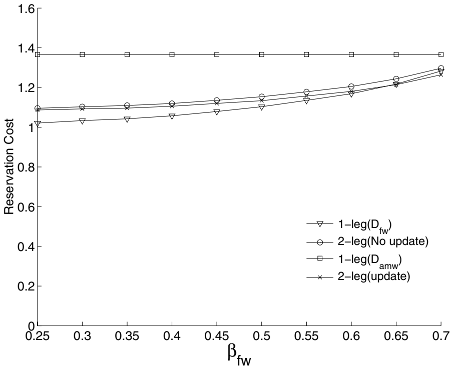


Fig. 7. Comparisons of user utility for different full warranty periods.

Fig. 8. Comparisons of reservation cost for different full warranty periods.

5.3 Update of Service Duration

In this set of experiments, we study the effect of at different times the service extension request is submitted in the Two-Leg reservation scheme in improving service continuity or user utility for those connections lasting longer than the initial full warranty period. In Fig. 9, we can see that both service continuity and utility do not change much when updates are submitted during the first 60% of the nominal service duration. After that, both increase. Figure 10 shows that reservation cost, has not much changes for different update submission times.

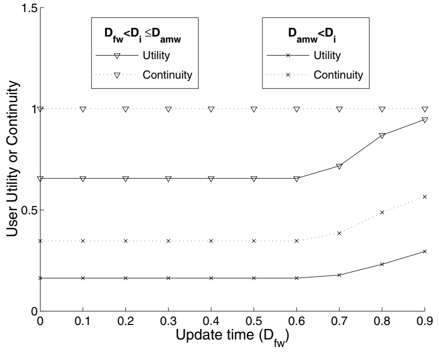


Fig. 9. Comparisons of service continuity and utility for service duration update of different submission times.

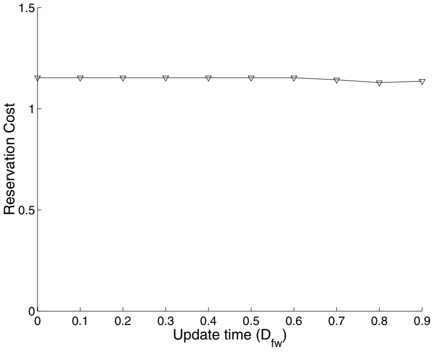


Fig. 10. The reservation cost for service duration update of different submission times.

6 Conclusion

It is difficult to do efficient resource management for advance reservations with uncertain service duration. In this paper we have presented a Two-Leg bandwidth reservation and admission control scheme. The idea is to perform bandwidth reservation in multiple stages. Each stage has a fixed duration and specific level of quality of service to assure. Thus, service provider can efficiently manage and allocate bandwidth needed to guarantee service quality to the connections at individual stages. Under the scheme, bandwidth reserved to an admitted AR request with uncertain duration includes a full bandwidth reservation for initial the service warranty period (Leg-One) and at least minimum bandwidth as well reserved for the after-warranty period (Leg-Two). The focus of the proposed scheme is not only to address the admission control issue at the initial call setup time but also the continuity of the service when events like overtimes occur.

An update mechanism is used to allow service user to update the network, especially a service duration extension is requested. If an update request cannot be satisfied, instead of reject a duration that best matches user's requirement is selected. In the worst case, Leg-Two bandwidth reservation assures at least minimum amount of bandwidth available to a connection to carry on the service.

The proposed scheme aims to provide service users a more predictive, affirmative service guarantees than gradually degrading service.

Finally, simulations are performed to evaluate the proposed schemes. Results show that the proposed scheme makes a good use of the bandwidth and outperform traditional one-time reservation in service continuity and user utility. The reservation cost is minimum, close to one-time reservation with fixed duration equal to the full warranty period even with additional bandwidth reserved for the at least minimum warranty period.

References

1. R. Braden, L. Zhang, "Resource ReSerVation protocol (RSVP) - Version 1: Functional specification, RFC 2205, 1997
2. W. Reinhardt, "Advance reservation of network resources for multimedia applications," in Proceedings of 2nd Intl. Workshop on Advanced Teleservices and High-Speed Communication Architectures (IWACA'94), Heidelberg, Germany, Sep. 1994.
3. A. Schill, F. Breiter, and S. Kahn, "Design and evaluation of an advance resource reservation protocol on top of RSVP," in Proceedings of IFIP Broadband'98, Stuttgart, Germany, April 1998.
4. F. Breiter, S. Kuhn, E. Robles and A. Schill, "The Usage of Advance Reservation Mechanisms in Distributed Multimedia Applicationsm," in Computer Networks and ISDN Systems, 30(16-18), Sept. 1998, pp. 1627-1635.
5. D. Ferrari, A. Gupta and G. Ventre, "Distributed Advance Reservation of Real-Time Connections," in NOSSDAV'95, Durhum, USA, Springer LNCS 1018, April 1995.
6. A. G. Greenberg, R. Srikant, and W. Whitt, "Resource Sharing for Book-Ahead and Instantaneous-Quest Calls," in IEEE/ACM Transactions on Networking, 1(7), April 1999.
7. M. Karsten, N. Berier, L. Wolf and R. Steinmetz, "A Policy-Based Service Specification for Resource Reservation in Advance," in Proc. of ICCC'99, Tokyo, Japan, 1999.
8. M. Degermark, T. Kohler, S. Pink and O. Schelen, "Advance Reservations for Predictive Service," in Proceedings of NOSSDAV' 95, Durham, NC, April 1995.
9. L. Delgrossi and L. Berger (Eds.), "Internet Stream protocol version 2 (ST2), protocol specification - version ST2+," in RFC 1819, Internet Engineering Task Force, August 1995.
10. R. A. Guerin and A. Orda, "Networks With Advance Reservations: The Routing Perspective," in Proceedings of INFOCOM 2000.
11. D. Wischik and A. G. Greenberg, "Admission Control for Booking Ahead Shared Resources," in Proceedings of INFOCOM'98 San Francisco, CA, April 1998.
12. The Blockbusters web page. http://www.blockbuster.com/
13. G. de Veciana, G. Kesidis and J. Walrand, "Resource Management in Wide-Area ATM Networks Using Effective Bandwidths", IEEE JSAC, Vol.13, No. 6, August 1995.
14. R. Guerin, H. Ahmadi and M. Naghshineh, " Equivalent Capacity and Its Application to Bandwidth Allocation in High-Speed Networks", IEEE JSAC, Vol. 9, No. 7, September 1991.

Performance Evaluation of the Deadline Credit Scheduling Algorithm for Soft-Real-Time Applications in Distributed Video-on-Demand Systems

Adamantia Alexandraki and Michael Paterakis

Technical University of Crete
Department of Electronics & Computer Engineering
Laboratory of Information & Computer Networks
GR-731-00 Chania, Greece
{ adalex,pateraki }@telecom.tuc.gr

Abstract. In this work we investigate a promising scheduling algorithm referred to as the Deadline Credit (DC) algorithm, which exploits the available bandwidth and buffer space in communication networks to serve a diverse class of prerecorded video applications. We provide simulation results when the DC algorithm is applied to a hierarchical architecture distributed VoD network, which fits the existing tree topology used in today's cable TV systems. The issues investigated via the simulations are: the system utilization, the influence of the buffer space on the delivered QoS, and the fairness of the scheduling mechanism. We examine cases with homogenous and diverse video streams. We also contribute a modification to the DC algorithm so that in cases when the video applications have different displaying periods, the video streams obtain a fair share of the network's resources. Finally, we validate our results by simulating actual videos encoded in MPEG-4 and H.263 formats.

1. Introduction

Telecommunication networks have lately extended to support a variety of services to the users such as tele-education, remote working, high-definition TV, web video streaming, etc. The demand of all these applications led to the concept of ***broadband integrated digital networks (B-ISDN)***, which are systems of high capabilities. The role of the ***scheduling process*** in B-ISDNs is to select each time the application that should be served next, so that the efficiency of the network is increased and the Q*uality of Services (QoS)* is kept at a satisfactory level.

Video-on-Demand (VoD) applications, which is the focus of this paper, can be classified somewhere between real and non-real time applications and they are usually referred as ***soft real-time applications***. When video is displayed, certain time constraints need to be met, i.e. the consecutive scenes in the video should reach the receiving end during finite time intervals. This brings video closer to real time applications. However, it is not necessary that the video scenes are generated in real time; they can be recorded *a priori*, but this entails the need for adequate buffer space. In this work we use VBR-encoded video, instead of CBR video, because for the same

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 202-213, 2002.

image quality VBR-encoded video can have a significantly lower average bit rate as well as exhibit considerable multiplexing gain.

We use the Asynchronous Transfer Mode (ATM) technique of switching and multiplexing multimedia streams, which encapsulates data into cells that travel along the network's links [4], [19]. A video application provides the network with a number of video frames at a rate that is referred as *frame rate*. This is the rate at which an image is digitized, compressed and cut to fit into ATM cells. It is also the rate at which frames should be available at the receiving end for the decoding and reconstruction processes. This periodicity as well as the video's frame size variability, should be taken into consideration in order to guarantee the arrival of frames within time constraints and the simultaneous avoidance of overflow at the receiving points.

We evaluate through simulation the performance of an innovative scheduling scheme referred to as the Deadline Credit Algorithm, originally proposed in [1] and [3]. We use the term ***application data unit*** (***ADU***), which refers to the ATM cells that belong to the same video frame. Each ADU has a sequence number (SN) indicating the scene that it belongs to. As a metric for performance evaluation the authors in [1] use the ***application data unit loss rate*** instead of the *cell loss rate,* since it is assumed that an image is degraded even if one of its cells is lost. We examine the DC algorithm and the role that buffers can play in increasing its efficiency. We consider only cases with small buffer availability because we intend to show that DC works quite well even when storage is a restricted parameter. Finally, we investigate the node utilization of the system, and we concentrate on evaluating the fairness of the scheduling process.

In section 1.1, we provide a brief overview of the DC algorithm. In section 2, we describe the distributed VoD system to be used in our study. Section 3 contains the simulation results for this system when the DC algorithm is applied for homogenous as well as diverse video streams. In section 4, we present simulation results when actual prerecorded MPEG-4 and H.263 video programs are used, and finally, in section 5, we present the conclusions of our work.

1.1 DC Algorithm Overview

The time within which an ADU has to reach its destination node can be spent in a variety of ways on the traversing nodes. The DC algorithm distributes this time over the nodes that an ADU traverses and poses restrictions on the time of departure from these nodes. Particularly, it poses an upper bound on the time when an ADU from stream j should have left the corresponding node in order to arrive at the next node within time constraints. Consequently, if the ADU cannot arrive in time at the next node, it is dropped, whereas, if an ADU is served, it is always guaranteed that it will arrive at the next node in time. The deadline is defined cumulatively from one node to the next, and if an ADU leaves a node before its deadline, the remaining time slots of its deadline can be spent at the next nodes.

Each time a stream should be selected for transmission, the DC algorithm selects the stream with the maximum number of ADU losses. Among streams with the same number of ADU losses the DC algorithm selects the stream with the minimum allowable delay, i.e. it serves the ADU that can be delayed the minimum time at the corresponding node. This is expressed in the priority index counter (P_CR), which is used to select between the contending streams the stream that should be served next. Let the superscript *j* denote the stream and the subscript *n* denote the node. Each stream at each node has a P_CR defined as follows:

$$\text{P\_CR}_n^j = \frac{D\_CR_n^j - T^j * L\_CR_n^j}{T^j} \tag{1}$$

where, the losses counter (L _CR$_n^j$) keeps information on the number of ADU losses that stream j has suffered from previous nodes up to node *n* and the deadline credit counter (D_CR$_n^j$, measured in slots) expresses the time that the ADU at the head of the queue of the stream *j* can be delayed at node *n*. The term T^j refers to the period of the stream. The smaller the value of the priority index counter of a stream, the higher the priority of the stream with respect to that of the other streams.

However, the buffer space at the traversing nodes is limited so before an ADU is transmitted the algorithm has to guarantee that there will not be buffer overflow at down stream nodes. In this work, we avoid buffer overflow using the buffer counter (Buffer$_n^j$)[1]. Buffer$_n^j$ is initialised one unit at node *n* for every ADU of maximum length from stream *j* it can hold. Whenever an ADU of stream j arrives at node n or whenever the transmission of an ADU from stream j, already stored in the corresponding buffer at node n, starts, the buffer counter is decreased or increased by one, respectively. Although, feedback for the exact free space of each node could be sent, we chose for simplicity to treat all ADUs as if they were of the maximum length. We assume that node *n* informs node *n-1* about its buffer condition, in order that node *n-1* does not transmit an ADU that will cause buffer overflow at node *n*. In contrast to [1], where the feedback is available periodically, in this work it is assumed that feedback is sent when an ADU is transmitted and the buffer occupancy changes.

2. Video-on-Demand System

Distributed information systems are able to support several kinds of applications. They are widely used for data, voice and video transmission. Particularly, for video applications, the case we are interested in, several network architectures have been developed that enable users to have access to a wide range of video programs on demand. In the following paragraphs we examine a hierarchical architecture for a VoD distributed network, which fits the existing tree topology used in today's cable TV systems [2], [18], [20], [21].

The distribution network topology is considered to be a binary tree associating a number of video servers and switching nodes connected via communication links[2]. At each node N$_{i,j}$ there is a storage device where some prerecorded video streams X$_{i,j,k}$ are stored, and a video server VS$_{i,j}$, which supplies children nodes with the video streams that are prerecorded at node N$_{i,j}$, as well as with those that arrive at N$_{i,j}$ from its father

[1] The original DC algorithm uses an upper bound on the D_CR counter to avoid buffer overflow. However, we found that this part of the algorithm does not function properly, therefore, the introduction of the buffer counter.

[2] We assume that the tree network is of binary form, although, in practice, the location of the distributed servers may differ.

stream node. A node selects a stream for transmission, according to a scheduling process (DC algorithm in our experiments), and starts transmitting it to the corresponding children node.

All the nodes of the tree network except those that belong to the last level run the DC algorithm for the streams they serve. The nodes of the last level, consume the arriving ADUs as well as the ADUs that are prerecorded at these nodes. At each system node there is also some buffer space for the arriving streams, where they are stored until they can be transmitted by the node. This buffer space is supposed to be limited and its size is a parameter of the system. Each arriving stream is supposed to have its own available buffer space**;** for the DC algorithm, this is denoted by the parameter Buffer_n^j for the stream j at node n.

All the users are connected to the leaf nodes, also called headends. When users require a video program, they connect to the appropriate headend and a request is sent cumulatively upwards through the traversing nodes until it reaches the source node. As soon as the source is informed, the connection is established and the video stream starts to travel along the appropriate path to reach the destination node.

Although there are many options relatively to the number of video programs stored at each node, in this work, we have chosen that all the network nodes have an equal number of prerecorded video programs (i.e. $X_{i,j}= \mathbf{X}, \forall$ i, j). **X** is another parameter of the system in the experiments we carry out. Furthermore, in our experiments all the prerecorded streams are considered “active”. This means that the level a node belongs determines the number of streams it serves, i.e. down stream nodes have more streams to serve than up stream nodes.

Finally, we assume that the users’ requests are uniformly distributed among video programs and that each video stream has been generated by a single headend request. Specifically, we assume that the prerecorded video streams of a node as well as the streams that arrive from its father node are distributed equally to its children nodes with the intention that the nodes of the same level are similarly loaded. The number of the served streams increases as we move down the tree, thus the nearer a network node to the destination nodes, the more loaded the node will be.

In the following experiments we assume a distributed video-on-demand system of depth three. The results shown below have been computed as average values from at least 10 replicated experiments (Monte Carlo Simulation). In each case, the simulations were of sufficient duration for the system to attain a steady state.

3. Performance Evaluation Experiments

3.1 Scheduling Homogenous Streams

The aim of the first experiment is to investigate how the DC algorithm works for a distributed system when homogenous streams are served (i.e. streams with the same periods). The main conclusion that we drew (results are omitted due to lack of space) is that the DC algorithm is fair. All streams suffer the same number of losses independently of their source-node location and the number of hops that they have to traverse to reach destination node. The DC algorithm guarantees that the quality of video delivered to the users, depends only on the system's capacity and not on the fact that some streams may monopolize the existing resources against the remaining streams.

3.2 Scheduling Streams with Different Frame Rates

In modern video-on-demand networks a diverse class of video traffic streams with different requirements is integrated. There is a wide variation in the volume of information generated when encoding different signals. Consider for instance two extremes types of video signals: Teleconference video images, on one hand, are nearly always images of people talking while directly facing the camera, usually viewed only from the waist up, and scene changes occur only rarely. These applications have uniform-activity-level; the change in the information content of consecutive frames is not significant. With broadcast media, on the other hand, there are no limitations on the content, and scene changes can be frequent. There are videos with sudden scene alteration and distinct changes in the subsequent scenes[3]. Uniform-activity-level applications can accomplish an acceptable quality even through a low frame rate, thus they are typically served with a frame rate equal to 10 or 15 frames/second. Broadcast television, on the contrary, requires higher bit rate transmissions for the same quality, on average, equal to 25 or 30 frames/second. As frame rate determines the period of an application, the incorporation of applications with various periods is frequent. In this section we examine the case when the multiplexed streams have different requirements relatively to their frame rate.

3.2.1 Experiment

All the traffic streams are assumed to have uniform length distribution in the interval [1,5]. There are **X** prerecorded streams stored at each node. We assume that a portion of 1/3 of them has period T=50 slots, another 1/3 has period T=100 slots and the remaining 1/3 has period T=200 slots. We simulated the system for **X**=12, 24, 36, 48 and buffer=1, 2 and 5 ADUs for each stream at each node. Each time the same number of streams with different periods reaches destination nodes. That selection was made so that all nodes are evenly loaded. The simulation ends when 300,000 ADUs from the stream with the highest period (T=200 slots) are required at destination nodes

The distribution of losses across video streams as observed from the second level of the tree network is shown in Fig. 1. The y-axis denotes the ratio of ADU losses and the x-axis the stream index. When stream index mod 3 is equal to 0 the stream has a period T=50 slots, if it is equal to 1 we have a stream with T=100 slots, and, finally, if it is equal to 2 it corresponds to a stream with T=200 slots.

The results are rather disappointing as far as fairness is concerned. Simulation demonstrated that the DC algorithm tries to distribute the number of losses uniformly across streams. However, as the periods of the streams differ, the streams suffer a

[3] Action movies, Formula 1 racing events and football matches are some examples of non-uniform activity-level videos.

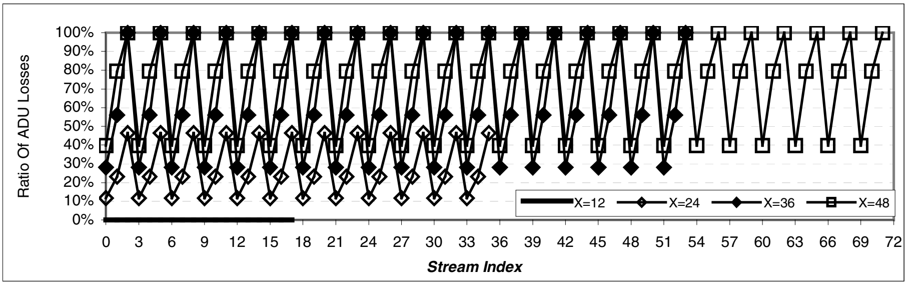


Fig. 1. Loss Distribution across streams with different periods

different ratio of losses. Particularly for large X, the streams with long periods undergo much more losses than the other streams.

The philosophy of the DC algorithm, as stated in [1], is to schedule from streams the one that has the smaller delay tolerance for the ADU at the head of the queue to expire. This favors streams with small period. For these streams the L_CR counters are updated faster when the network is overutilized; therefore they are selected more frequently. The streams with higher periods are neglected, especially in cases when a large number of streams are multiplexed and many ADU losses occur. A lost ADU is taken into account in the update of the P_CR counter independently of the stream's period; however, an elapsed slot acts in favor of streams with small period compared to streams with higher period. At times of network's over-utilization, this acts against the streams with higher periods, which are seldom selected and their loss counters are infrequently updated.

3.2.2 A Proposed Modification of the DC Algorithm

The video quality that the end user perceives depends on the application's period. Even if we achieve to distribute the number of ADU losses uniformly among streams, we would have to attain an equal ratio of losses among streams in order to accomplish the same video quality. For a stream with long period, losing an ADU corresponds to a greater timing interval that the video is degraded compared to losing an ADU from a stream with a smaller period. As a result, if the goal is that all the streams independently of their periods achieve the same video quality, the DC algorithm should be modified. The goal should be that all streams experience an equal portion of ADU transmissions as well as an equal portion of ADU losses. Let the $data_n^j$ correspond to the number of ADUs that were transmitted by node n from stream j. Then, if instead of the relation (1), as in [1], we use the relation

$$\mathbf{P\_CR}_n^j = \mathbf{D\_CR}_n^j + \mathbf{data}_n^j * \mathbf{T}^j - \mathbf{L\_CR}_n^j * \mathbf{T}^j \qquad (2)$$

then among streams with the same data and L_CR values the one with the smallest deadline allowance would be selected. Since $D\_CR_n^j$ counters are increased T^j units for every transmission or loss [1], the streams with small periods will be served first because for the same number of transmissions (data) and losses (L_CR), they have smaller D_CR value. After these streams are served, their data and L_CR counters will be updated, thus in the next examination slot a stream with long period will be selected. Among streams with different data values but equal D_CR and L_CR values, the stream with the least transmissions will be chosen. Likewise, among streams with the same data and D_CR values, the one with the most losses will be chosen to transmit next.

Equation (2) attempts to balance the number of losses and transmissions among streams. It works in accordance to (1), when streams have the same period, since for the same losses and transmissions, D_CR plays the primary role. Additionally, equation (2) eliminates the problem illustrated in the previous section as each time a transmission occurs $data_n^j$ is increased by 1. The selection of a stream now depends not only on the losses it has suffered but also on the total number of transmitted ADUs. An elapsed slot plays the same role independently of the stream's period, while a lost or transmitted ADU influences the priority counter in proportion to the streams period.

Generally, equation (2) expresses the total time that a stream is ahead or behind. Expanding the relation (2), the term $data_n^j * T^j$ expresses the time that the end user receives faultless video, while the term $L\_CR_n^j * T^j$ expresses the time that the end user experiences video quality degradation. The $D\_CR_n^j$ term is used so that among streams with the same transmissions and losses, the one with the shortest time till expiration is selected. Generally, the relation (2) attempts to provide all the streams with the same video quality in a time-oriented manner. The main idea is that fairness will be achieved if all the users receive the same times of faultless and degraded video, respectively.

3.2.3 Simulation Results

The results we obtained from the simulation of the modified DC algorithm are shown in Fig. 2. We simulated exactly the same experiment with the one in 3.2.1 and obtained precisely the same total number of ADU losses. This is expected, since the total number of ADU losses depends exclusively on the load of the system. The difference arises on the dispersal of losses among the contending streams. Notice the absolute uniform distribution of the ratio of ADU losses. All streams independently of their period undergo the same ratio of ADU losses, thus all viewers observe the same video quality.

4. Performance Evaluation of DC Algorithm for MPEG-4, H.263 Video Traces

In real videos, there are some dependencies between the consecutive frames as well as a non-uniform frame length distribution. The scope of the following experiments is to extend the conclusions drawn from the results of the previous section to the case of actual video traces.

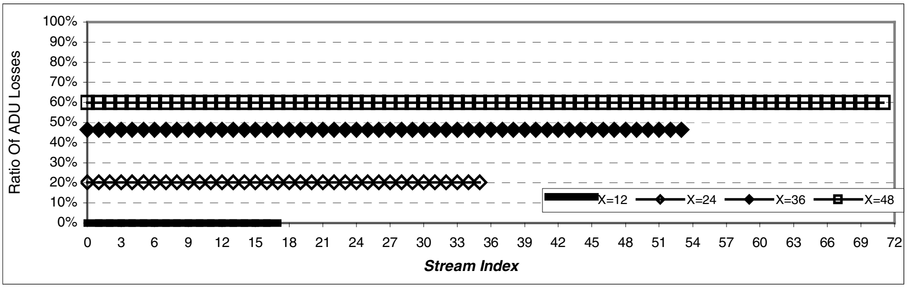


Fig. 2. Loss distribution across streams with different periods for the modified DC algorithm

4.1 MPEG-4 Videos

In the following experiments we use MPEG-4 video trace files originating from existing video programs. The video programs we have chosen to use in our experiments cover a wide range of video applications such as action movies, cartoons, tele-conference, ski, formula 1 racing events, music video clips, talk shows, etc. Each video is encoded at three different quality levels: high, medium and low quality[4].

The Group of Pictures (GoP) pattern was set to I B B P B B P B B P B B. Due to the dependencies between frames not all the frames have the same significance relatively to the image that the end viewer actually perceives. I frames are vital for all the subsequent frames of the group to be displayed, so if an I-type frame is lost, then the whole group is considered to be lost and none of the remaining frames is transmitted. Additionally, if a P-type frame is lost, the rest of the group is considered to be lost and the subsequent frames are not transmitted. Finally, a B-type lost frame accounts for a single loss, since no frame depends on B frames.

4.2 Experiment with MPEG-4 Videos

We simulate approximately one hour of operation for the binary tree network for **X**=4, 8, 12, 16, 20, 24 and 28 programs, and buffer=1, 2 and 5 ADUs of the maximum length. We present the simulation results for a bandwidth of **12.72 Mbps** for high quality video[5].

Our simulations have shown that in each of the three cases examined, the DC algorithm distributes fairly the losses across the multiplexed video streams and gives almost the same ratios of I-, P- and B-frame losses to the multiplexing streams. Notice in Fig. 3 the agreement in the ratios of I-, P- and B-frame losses among the multiplexed video streams (i.e. streams of the first and second level of the tree). This is another indication of the fairness of the DC algorithm.

[4] For an extended description of the videos we used, the read is referred to [15].

[5] Simulation results are also available for medium and low quality video, but are omitted due to space constraints.

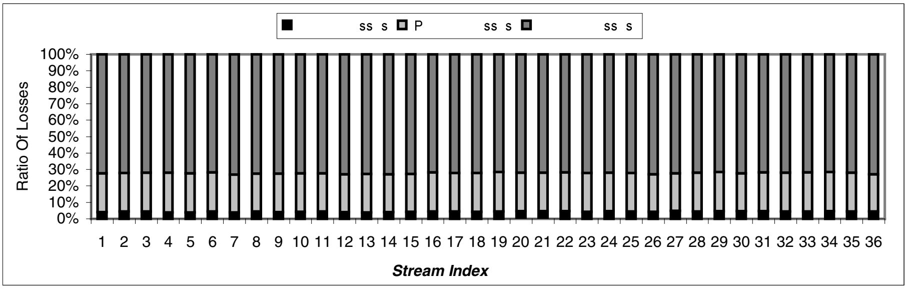


Fig. 3. Ratio of I-, P-, B-frame losses for Buffer=1 ADU, X=12, Bandwidth=12.72 Mbps

Another interesting conclusion that can be drawn is that the placement of the video programs can be chosen independently of the video programs' popularity. What usually happens is that the video program's popularity determines its placement in order that the most popular programs are stored closer to the users, whereas the least popular ones are stored at the upper levels of the network tree topology [2],[18]. This choice is made so that the most popular videos are located closer to the users, and suffer the least start up delay and the least number of frame losses. However, our experiments have shown that all streams independently of the level at which they are stored suffer the same number of losses. Based on this observation, we can claim that DC algorithm also gives flexibility to the system configuration regarding the placement of the video programs.

In our experiments we have assumed that each program is required only by one children stream node. In a real system, some videos may be required by both of a node's children stream nodes, while others may not be active. As revealed by our experiment, the way the DC algorithm distributes the losses across the multiplexed streams depends exclusively on the number of multiplexed streams and the available bandwidth. Since the DC algorithm distributes the losses fairly across the multiplexed video streams independently of the videos' contents, we can assume that the DC algorithm is also fair when some videos are required by more than one headends. However, notice that the location of content still has a potential impact on network utilization and user startup latency.

4.3 H.263 Videos

In the following experiment we have used H.263 video compression with variable bit rate, too. We enable PB-frames, also called "stuffed" frames. A PB-frame consists of two consecutive frames that are encoded as one unit [15].

H.263 encoding produces I (usually only the first frame is an I-fame), P and PB frames. I and P frames have a period equal to 40msecs, whereas PB frames have a period equal to 80msecs. Consequently, H.263 may produce variable frame periods. Videos with plenty of information generate, except of PB-frames, many I- and P-frames, thus they have often period-alterations between 40 and 80 msecs arising from the frames generated. On the contrary, videos with low scene alteration and uniform level activity (such as Lecture Room Camera, Office Camera etc) generate almost exclusively PB-frames, leading to an unchanged period equal to 80 msecs.

4.4 Experiment with MPEG-4 and H.263 Videos

The purpose of this experiment is to test and further support the modification we proposed in section 3.2.2 relatively to the fairness of the DC algorithm when video applications with different frame rates are supported in the same network At each node they are stored **X** prerecorded video streams that supply the children nodes with traffic. **X/2** streams are high, medium and low quality MPEG-4 encoded and the remaining **X/2** are H.263 encoded. For the purpose of our experiment, we have chosen only programs with a few scenes alterations (e.g. video conference programs) that contain information, which can be presented exclusively with PB frames. Consequently, from the MPEG-4 encoding video programs the transmission of one frame per 40 msecs will be required, while from the H.263 encoding video programs the transmission of one "stuffed" frame per 80 msecs will be required. The available bandwidth is assumed equal to **9.54 Mbps.**

Fig. 4 shows the simulation results when the P_CR is defined as in [1]. Streams with odd index numbers refer to MPEG-4 video programs, while streams with even index numbers refer to H.263 video programs. Notice that the ratio of ADU losses depends on the period of the stream. In contrast, for exactly the same experiment, as shown in Fig. 5 the modified DC algorithm gives a uniform distribution of the losses across multiplexed streams.

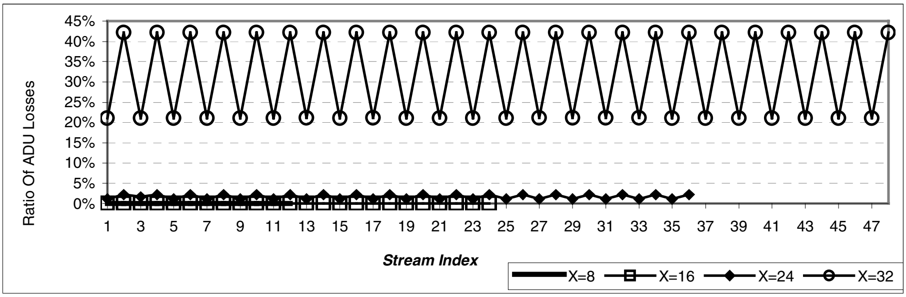


Fig. 4. Loss Distribution across MPEG4 and H.263 video programs according to [1]

5. Conclusions

The DC algorithm uses dynamic allocation of bandwidth and distributes the available bandwidth fairly between all the competing video streams. When the system is under-utilized, the DC algorithm forwards an equal number of ADUs from each traffic flow at down stream nodes and keeps the streams at the same level of delay. On the other hand, when the network is over-utilized and frames need to be dropped, DC distributes the losses fairly across streams. The efficiency of the DC algorithm with regard to the ADU losses increases as the available buffer space at the traversing nodes increases, because the DC algorithm exploits the available buffer space to send ADUs in advance.

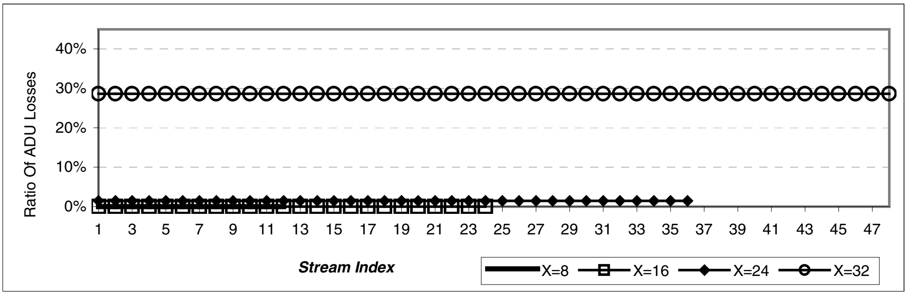


Fig. 5. Loss Distribution across MPEG4 and H.263 video programs for the modified DC algorithm

We evaluated the performance of the DC algorithm for a distributed Video-on-Demand network, which fits the existing tree topology used in today's cable TV systems, and validate our results by simulating actual videos encoded in MPEG-4 and H.263 format. The main conclusion we drew is that the DC algorithm is fair and distributes losses evenly across multiplexed streams. This means that all streams suffer the same number of losses independently of their source-node location. This can contribute to a flexible placement of the video programs, since the placement can be chosen independently of the video programs' popularity.

Finally, we provided simulation results that support the modification we proposed. The conclusion is that the modified DC algorithm treats fairly videos with different frame rates, with different quality encodings and with different encoding methods, i.e. a diverse class of video applications can be integrated into the same network and all these applications are treated fairly.

References

[1] Z. Antoniou, I. Stavrakakis, "Efficient End-to-End Transport of Soft Real-Time Applications", pp. 470-482, IFIP Networking'2000, May 14-19, 2000, Paris, France

[2] Constantinos C. Vassilakis, "Modeling, Design and Performance Evaluation of Interactive Distributed Video-On-Demand Systems", M.Sc. Thesis, ECE Department, Technical University of Crete, 1999. Part of this M.Sc. Thesis has been published in the ACM Multimedia Systems Journal, Vol. 8, No. 2, 2000, pp 92-104, under the title "Video Placement and Configuration of Distributed Video Servers on Cable TV Networks", (authors: C. Vassilakis, M. Paterakis and P. Triantafillou).

[3] Z. Antoniou, I. Stavrakakis, "Deadline Credit Scheduling Policy for Prerecorded Sources", IEEE GLOBECOM'99, Dec. 5-9, 1999, Rio, Brazil.

[4] R. Onvural,"Asynchronous Transfer Mode Networks - Performance Issues", Artech House, 1995

[5] UYLESS D. BLACK, (Second Edition), "Data Communications And Distributed Systems", Prentice-Hall International Editions, 1987

[6] Daniel Minoli, "Video Dialtone Technology", McGraw-Hill, Inc, 1995

[7] Naohitsa Ohta, "Packet Video Modeling and Signal Processing", Artech House, 1994

[8] Rafael C. Gonzalez, Richard E.Woods, "Digital Image Processing", Addison-Wesley Publishing Company,1992

[9] R. J. Clarke, “Digital Compression of still images and video”, Academic Press, 1995

[10] Moving Pictures Expert Group Organization, http://www.mpeg.org

[11] P. Cherriman, L. Hanzo and R. Lucas, ARQ-assisted H261 and H263-based Programmable Video Transceivers, 1995/6 Research Journal, Communications Group, University of Southampton, http://www.ecs.soton.ac.uk/publications/rj/1995-1996/comms/pjc94r/rj95.htm

[12] H261 Video Coding, http://www-mobile.ecs.soton.ac.uk /peter/h261 /h261.html

[13] H263 Video Coding, http://www-mobile.ecs.soton.ac.uk/peter/h263/h263.html

[14] H261 Overview, http://www.nyquist-media.co.uk/streaming/h261.html

[15] Frank H.P. Fitzek and Martin Reisslein, “MPEG-4 and H.263 Video Traces for Network Performance Evaluation”, Technical University Berlin, Telecommunication Network Group, TKN Technical Report Series, TKN-00-06, October 2000, http://www-tkn.ee.tu-berlin.de/research/trace/trace.html

[16] Joan Mitchell, William B. Pennebaker, Chad E. Fogg, and Didier J. LeGall, “MPEG Video Compression Standard”, International Thomson Publishing, 1997

[17] Jean-Pierre Leduc, Digital Moving Pictures – Coding and Transmission on ATM Networks, Volume 3, Elsevier, 1994

[18] C. Bisdikian and B. V. Patel, “Issues on Movie Allocation in Distributed Video-on-Demand Systems”, IEEE International Conf. on Communications, 1995, pp 250-255. Also published in IEEE Multimedia Magazine under the title “Cost-Based Program Allocation for Distributed Multimedia-on-Demand Systems”, IEEE Multimedia Magazine, Vol. 3, No. 3, Fall 1996

[19] ATM Forum, http://www.atmforum.com

[20] J.P. Nussbaumer, B. V. Patel, F. Schaffa, and J. P. G. Sterbenz, “Network Requirements for Interactive Video on Demand”, IEEE Journal on Selected Areas in Communications, Vol. 13, No 5, June 1995.

[21] G. Bianchi, R. Melen, “ Performance and Dimensioning of a Hierarchical Video Storage Network for Interactive Video Servers”, European Transactions on Telecommunications, Vol 7, No. 4, July-August 1996

The Impact of Replacement Granularity on Video Caching

Elias Balafoutis, Antonis Panagakis, Nikolaos Laoutaris, and
Ioannis Stavrakakis

Department of Informatics & Telecommunications,
University of Athens, 15784 Athens, Greece
{balaf,apan,laoutaris,istavrak}@di.uoa.gr

Abstract. In this paper the idea that large objects, such as video files, should not be cached or replaced in their entirety, but rather be partitioned in chunks and replacement decisions be applied at the chunk level is examined. It is shown, that a higher byte hit ratio (BHR) can be achieved through partial replacement. The price paid for the improved BHR performance is that the replacement algorithm, e.g. LRU, takes a longer time to induce the steady state BHR. It is demonstrated that this problem could be addressed by a hybrid caching scheme that employs variable sized chunks; the use of small chunks leads to the maximization of BHR in periods of stable video popularity, while large chunks are used when extreme popularity changes occur to assist the fast convergence to the new steady state BHR.

1 Introduction

The explosive growth of demand for bandwidth, fuelled by the introduction of the world wide web in the early nineties, found data networks unprepared to handle the new traffic volumes. This led to an increase in both loss rates and user perceived latency, that could easily hamper the new global information delivery system. Caching, i.e, replication of popular data objects close to the demanding clients, has been successfully used to relieve the backbone network and reduce the delivery delay of requested data. In a similar way contemporary networks, although copying adequately with web traffic, seem to have difficulty in managing effectively the delivery of information-rich content such as streaming video, which is rising as the new popular media to be integrated in the internet infrastructure. The large size of videos not only overloads data connections but also easily exhausts the capacity of conventional web caches.

A variety of caching schemes have been proposed to handle video. Initial works, have inherited the main characteristics of web caching schemes and have treated videos as single entities which are either cached completely or not at all [1,2,3]. More recent works take into consideration the special characteristics of videos : their structure, the associated rate variability, their large volume and the need for a time-constrained delivery of content. The later does not only refer to a requirement for a small initial delay, to preserve the interactivity of the

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 214–225, 2002.

service, but also to a requirement for an isochronous delivery of the media units (video frames) that make up a video stream.

In [4] the initial frames of each video (called the prefix) are cached in the proxy in order to improve the startup latency experienced by users. Additionally, smoothing is performed to reduce the peak bandwidth and increase the utilization in leased network channels that connect the proxy with the origin server. In a similar approach in [5], the bursty part of a VBR video stream is selected to be stored at the proxy while the remaining smooth part is retrieved directly from the repository, again reducing the peak bandwidth requirement in the backbone links. Both aforementioned schemes deal with the burstiness, which is inherent in VBR encoding algorithms.

In [6,7] the prefix is stored in the cache and the remaining part (the suffix) is either explicitly requested from the video repository or retrieved through an ongoing multicast transmission which services a group of concurrent users; in the later case, a patch might be requested directly from the repository, so as to fill the gap between the prefix and the currently multicasted part of the suffix. Request merging is also proposed in [8,9] in the form of window-based caching schemes. In particular, local proxies cache a sliding window of data, trying to merge requests for the same stream that arrive closely in time.

The caching of layered encoded video is studied in [10,11]. In [10] an optimization algorithm determines which videos and which layers should be cached. In [11] the focus is on the maximization of the perceived quality for popular videos that are delivered over best-effort networks.

Unlike traditional web caching, most of the above video caching schemes (most of them have been designed for video on demand systems) do not conform to the dynamic nature of caching according to which cache contents are dynamically updated in a demand driven fashion.

The performance of a proxy is characterized by its ability to reduce the amount of data that cross the backbone network (captured by the BHR), and also by the ability to provide streaming services with acceptable initial delay. The overall performance is sensitive to sudden changes in popularity. If the cache does not detect such changes quickly, there could be substantial mismatch between the content of the cache and the upcoming content requests, leading to a low hit ratio and an increased initial delay.

In what follows it is proposed that a video be segmented into a number of chunks and replacement decisions be taken at the chunk level rather than based on the entire video. This partial caching has a twofold beneficial effect: it increases the BHR compared to entire video caching; and it reduces the perceived delivery delay, as a significant number of initial parts may be found in the cache upon request. The existence of the initial part of a stream in the cache also allows for jitter concealment in the path from the proxy to the server. Our work studies the effect of the replacement granularity on cache performance and the trade-off that exists between cache performance and responsiveness to popularity changes. We show that the price paid for using partial caching is a slower reaction to popularity changes. We attempt to eleviate this drawback by using variable sized replacement units.

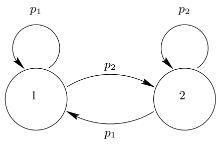


Fig. 1. State diagram of the cache for the first scenario(*).

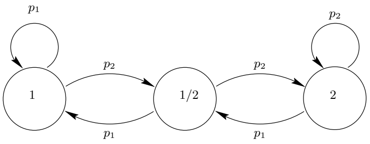


Fig. 2. State diagram of the cache for the second scenario(*).

(*)State 1 (state 2) corresponds to video 1 (video 2) being entirely cached and state 1/2 corresponds to the first half of each video being cached. The transition probabilities p_1 and p_2 are equal to the corresponding request probabilities.

2 Motivation of the Work – Intuitive Considerations

Cache replacement algorithms utilize the request history to estimate the current request probabilities and self-organize accordingly. Given a stationary request pattern and a large number of request samples, the underlying request probabilities can be "learned" by counting the requests for each video. Replacement algorithms are able to provide a good estimate of the request ranking without the need to count a large number of requests, e.g. LRU simply replaces the least recently used object upon a request of a new object.

In web caching the arrival of a single request changes slightly both the recent request history and the state of the cache, since the size of an ordinary web page is very small compared to the capacity of the cache. In video caching a single replacement causes a relatively greater change to the state of the cache, although a single request has similar impact on the recent request history as in web caching. Chunk-based replacement strategies (as studied here) try to establish a "Web-like" relation between a single request and the corresponding replacement unit.

The potential advantage of chunk-based replacement strategy is demonstrated in the following simplified example. Assume that there are two equally sized videos, competing for a place in the cache that can fit entirely only one video. Let p_1 (p_2) denote the probability that video 1 (video 2) is requested. In addition assume that a request-based replacement algorithm is used. A video that is not found in the cache upon request, is cached when it arrives from the server, taking up storage space that was held by the other video. Two scenarios are considered. In the first scenario the replacement unit is equal to the entire video, implying that videos are cached or removed from the cache entirely. The state of the cache upon replacement epochs[1] can be modelled as a two state Markov chain (depicted in Fig. 1). The second scenario allows the partial replacement of video, with a replacement unit that is equal to half of a video. When the requested video is completely or partially missing from the cache, one half of the previously cached video is flushed, and one half of the requested video is being cached. Replacement takes place in such a way that if a video is partially

[1] We assume that a video is immediately downloaded upon its request, so replacement decisions occur at request arrival instants.

cached, the cached part is always its first half. This implies that there are three possible states of a full cache, namely the first half of each video, the entire video 1, or the entire video 2 being cached. The state diagram of the cache content according to the second scenario is illustrated in Fig. 2.

For both scenarios, let the cost of a total cache miss (the entire requested video is missing from the cache) be equal to 1 and the cost of a partial cache miss (one half of the requested video is missing) be 0.5. If the requested video is completely cached, a cache hit occurs, and no cost is incurred. It is straightforward to calculate the steady state probabilities and costs for each scenario. More specifically, the steady state cost is equal to $\sum_{i,j} \pi_i P_{ij} c_{ij}$ where P_{ij} is the transition probability from state i to state j, c_{ij} is the cost corresponding to this transition and π_i is the steady state probability of state i. If C_1 (C_2) denotes the steady state cost for the first (second) scenario, then:

$$C_1 = 2p_1p_2 \qquad C_2 = \frac{3p_1p_2}{2(1-p_1p_2)} \tag{1}$$

which implies that $C_1 \geq C_2$ for all p_1 and p_2 (equality holds only for the special cases $p_1 = 0$, $p_1 = 1$, and $p_1 = .5$)[2].

3 System Description

3.1 Network Topology

Figure 3, illustrates the topology of the video dissemination system under consideration. Videos are stored at geographically dispersed origin servers. A proxy server is installed at the same local area network with a number of clients. Requests for videos are directed to the proxy which services them either from its local cache or by contacting the origin servers over the wide area network. The proxy caches the most popular videos trying to reduce the accesses to the servers and consequently the volume of data transmitted over the wide area network. It is assumed that there is abundant bandwidth between the proxy and the clients to support video streaming. On the other hand, the transmission of videos from the origin servers over the wide area network is expensive as it consumes bandwidth, which is a scarce resource in the backbone.

3.2 Proxy's Internal Architecture

Figure 4 illustrates the internal architecture of the porxy. The proxy consists of two major entities: the request manager and the cache manager. The request manager accepts all the client requests and is responsible for the continuous streaming of video towards the clients. In general, its responsibility is to schedule

[2] The aforementioned analysis was carried out under the assumption that request interarrivals are always greater than the time it takes to download half or the entire video; this in essence means that replacement decisions are implemented instantly and do not have to wait until the missing data arrive from the origin server.

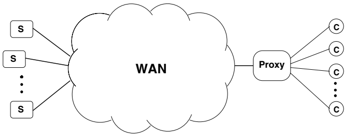


Fig. 3. Network topology: the origin servers (S) hold the available videos; clients (C) request videos and the proxy server services these request.

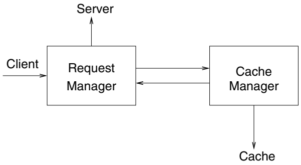


Fig. 4. Internal proxy architecture

both the transmission of the prefix to the clients. The main responsibility of the cache manager it to efficiently allocate the proxy storage resources to the requested videos. This work focuses on the functionality of the cache manager.

To allow for the isolated study of the cache manager, only a simple request manager is considered. It is assumed that the request manager immediately initiates the transmission of the prefix (if any) to the client and requests the suffix from the origin server. In addition, it is assumed that the suffix can be (and is) delivered exactly when it is needed from the origin server[3].

The cache manager receives incoming data from the origin server and decides whether the new data will be cached or not. When the decision is to cache the newly retrieved video parts the cache manager also decides a) how much space to dedicate to this video, b) which of the missing parts of the video to hold, and c) which data to remove from the cache to make room for the data.

In particular, the cache manager uses a fixed caching/replacement unit, called a chunk. When a video that is not in the cache is requested it is fetched from the server and its initial chunk is stored in the proxy. In case there is not sufficient space in the cache the replacement algorithm selects a video for removal. The last chunk of the selected video is removed. Each additional request for the same video results in the caching of an additional (consecutive) chunk. This guarantees that only the prefix (initial consecutive parts) of each video are cached. The objective is to investigate the impact of the replacement granularity on the overall performance of the system.

4 Performance Evaluation

4.1 Preliminaries

As mentioned in Sect. 3.2, the main responsibility of the cache manager is to efficiently manage the proxy's storage resources so as to reduce the volume of the data that are fetched from the origin servers. This performance aspect is captured by the Byte Hit Ratio (BHR), which is the fraction of data that can be

[3] A more advanced request manager could, for example, perform request batching to service nearby requests with a single connection to the client.

served directly from the cache's local storage. Here, the BHR for a single request for video i is defined as:

$$BHR_i = \frac{\text{Size of the cached portion of the requested video } i}{\text{Size of the complete video } i}$$

BHR_i takes values between 0 and 1; 0 for a complete miss, and 1 for a complete hit. The average BHR of all requested videos over an interval[4] x is denoted as $BHR(x)$. The steady state BHR (ss-BHR) is determined from $BHR(x)$ as x tends to infinity and assuming that no popularity changes occur, i.e. the request probability of a video is assumed to remain unchanged over the interval x.

The independent reference model [12] is assumed, according to which a video i is requested with probability p_i independently of previous requests. If the request probability of each video is known and assuming a unicast-only backbone in the path from the server to the proxy, it can be shown that the optimal caching policy – in terms of bytes that cross the backbone link – is the Highest Popularity First (HPF) policy. Under HPF, the proxy stores entire videos in descending order of popularity, until its cache capacity is reached. Only the last video is partially cached. The optimality of HPF stems from the fact that HPF is an optimal solution to the partial knapsack problem: maximize $\sum_{i=1}^{N} v_i \cdot p_i$ under the constraints: $\sum_{i=1}^{N} v_i \leq S$, $0 \leq v_i \leq L_i$, where L_i is the length of video i in number of chunks, v_i is the cached prefix of video i in number of chunks, S is the proxy's storage capacity in number of chunks, and N is the number of available videos (see [6] for details).

The performance of the proposed video caching scheme is evaluated via simulations. The main metric of interest is the BHR. The responsiveness of the cache to popularity changes is jointly considered.

4.2 Simulation Model

We have constructed a simulation model that consists of a video server with a set of N videos, and a proxy server with a storage capacity of K complete videos; the ratio K/N captures the relative cache size of the proxy. For simplicity it is assumed that all videos are constant bitrate encoded and are of equal length L units (under the constant bitrate assumption video length units can be time or storage units). p_i denotes the request probability of video i – it is also refered to as the popularity of video i – and follows a Zipf distribution, i.e., $p_i = C/i^a$, where $C = (\sum_{i=1}^{N} \frac{1}{i^a})^{-1}$. a is the Zipf parameter determining the skewness of the distribution. It is assumed that request arrivals follow a Poisson process with mean rate λ. The parameters of the simulation study are summarized in Table 1.

The popularity changes considered in the simulation, were implemented using the following setting: whenever a popularity change is about to occur, we transpose the popularities of videos, i.e., popular videos become unpopular and vise versa. Under the new popularity distribution, unpopular videos that were

[4] the interval x can be a time interval (e.g., a day) or a number of requests.

Table 1. Simulation parameters

Notation	*System Parameters*	*Default Values*
L	Video Size / Duration	1000 units / 1hour
K	Cache Size	100 videos
N	Number of Videos in the repository	1000
K/N	Relative Cache Size	0.1
a	Zipf parameter	0.8
λ	Mean request arrival rate	30 req/hour

missing from the cache appear as new hot videos and eventually capture a significant part of the cache. Previously popular videos are made unpopular and are eventually pushed out of the cache.

In our simulations the LRU replacement policy is used with a slight modification that prevents the replacement of chunks belonging to "active videos" (videos that are currently streamed), i.e., the least recently used inactive video is selected for the replacement. LRU was chosen as the most widely studied replacement policy which is often used as a comparison standard.

5 Simulation Results

Cache State. Figures 5 and 6 provide a visualization of the content of the cache as time evolves. For illustration purposes, only a total population of five videos and a cache capacity of three videos is considered. The cache is empty prior to the first request arrival and fills up as requests arrive. The LRU replacement policy is activated as soon as the total capacity is reached. The video size is assumed to be 1000 units. The results for chunk sizes of 2, and 100 units and a Zipf parameter $a = 0.8$ for video-1 to video-5 (descending popularity) are shown in Fig. 5 and 6 respectively. In Fig. 7 the optimal static allocation of the cache storage is illustrated. From Fig. 5 and Fig. 6 it becomes clear that for a small chunk size the cache state converges to the optimal static allocation slowly and with negligible oscillations, while for a greater chunk size, the cache state converges fast but significant oscillations appear. Oscillations are expected to

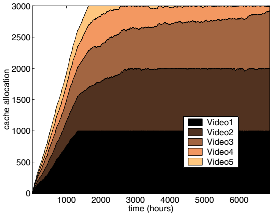


Fig. 5. Cache state for chunk size = 2 units

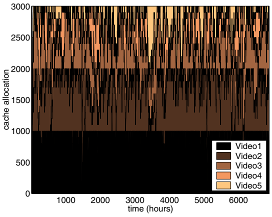


Fig. 6. Cache state for chunk size = 100 units

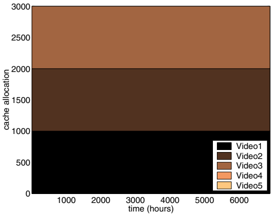


Fig. 7. Optimal static allocation

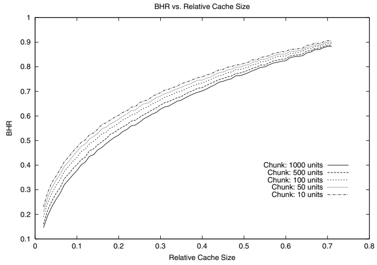


Fig. 8. BHR vs. Relative Cache Size. Zipf parameter 0.8

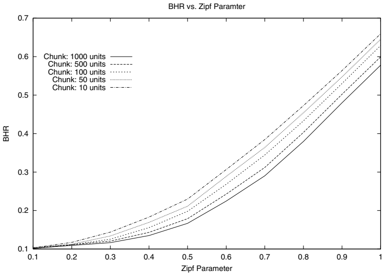


Fig. 9. BHR vs. Zipf parameter. Relative Cache Size 10%

have a negative impact on the BHR while the convergence time is expected to affect the capability of the system to adapt to popularity changes. The underlying tradeoff is investigated in detail in the sequel.

The Byte Hit Ratio. Fig. 8 illustrates the effect of the relative cache size on BHR, for several chunk sizes. BHR increases with the relative cache size since a greater number of chunks fit in the cache. Moreover, for a specific relative cache size, small chunks lead to a higher BHR. Similar observations apply to Fig. 9 that depicts the BHR as a function of the Zipf parameter, for different values of the chunk size.

Figures 10 and 11 [5] (for chunk sizes 10 and 1000 respectively) depict the BHR versus the sum of the popularities $\sum_{i=1}^{m} p_i$ of the videos that fit in the cache, when videos are placed in the cache according to descending popularity order; m is the index of the least popular video that fits in the cache. This sum depends on two factors: the size of the cache and the skewness of the popularity distribution[6]. Each line in Figures 10 and 11 corresponds to a different value of the Zipf parameter and the points of each line correspond to different values of the cache size. From the figures it follows that for a small chunk size different pairs of skewness and cache size result in the same BHR if the sum of the popularities of the videos that fit in the cache is the same. That is, the latter sum fully determines BHR under a small replacement unit. This conclusion suggests that a smaller cache size would be required to achieve a certain BHR if the video request probabilities are highly skewed, compared to the case under less skewed request probabilities. For a greater chunk size this result seems to hold only for zipf parameters greater than some value e.g 0.5.

The impact of the chunk size on BHR is illustrated in Fig. 12, for the system parameters presented in Table 1. For these parameters the sum of the popu-

[5] These figures relate to a discussion that is motivated by results in [13], where the relation between the fault probability of LRU and the tail of the popularity (request) distribution is demonstrated.

[6] For a specific cache size, this sum increases as the zipf parameter increases, since a greater request probability (p_i) is associated with the videos involved in the sum. For a specific value of the zipf parameter the sum increases as the cache size increases, since more videos are involved in the summation.

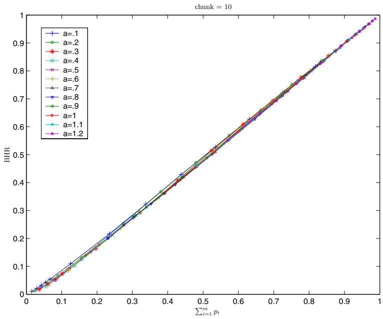


Fig. 10. BHR vs $\sum_{i=1}^{m} p_i$ for chunk = 10, where m is the index of the least popular video that fits in the cache

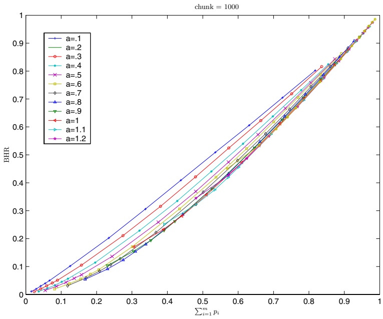


Fig. 11. BHR vs $\sum_{i=1}^{m} p_i$ for chunk = 1000, where m is the index of the least popular video that fits in the cache

larities of the videos that fit in the cache when videos are placed in the cache according to descending popularity order is 0.525. It is observed that as the chunk size increases, the BHR reduces initially fast and then slowly converges to the BHR achieved when complete videos are used as a replacement unit.

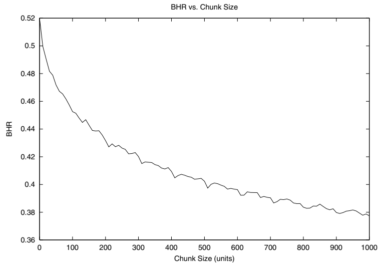


Fig. 12. BHR vs. Chunk Size, for Relative Cache Size 10% and Zipf parameter 0.8

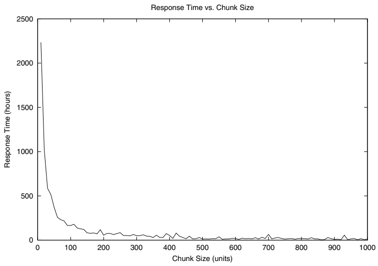


Fig. 13. BHR vs. Response Time, for Relative Cache Size 10% and Zipf parameter 0.8

Responsiveness. Upon a change of popularities, the BHR is expected to decrease for some period and then converge to the new steady-state value. It should be noted that the new ss-BHR is not necessarily the same with the old one as it depends on skweness of the popularity distribution. Responsiveness can be qualitatively defined as the ability of the system to adapt to changes in popularities. In order to quantitatively capture this performance aspect, we flush the cache[7] and measure the time needed for the BHR to reach 90% of its steady-state value. In Fig. 13 the response time is illustrated for several chunk sizes. As expected, the response time is small for large chunk sizes and increases rapidly as the chunk size decreases.

[7] This is an extreme case of popularity change since it is equivalent to a cache full with totally unpopular videos.

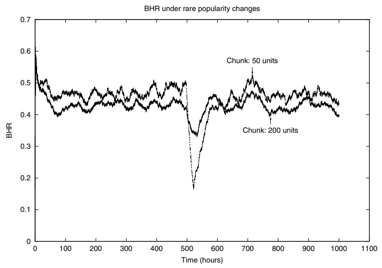


Fig. 14. Rare popularity changes. Average BHR for chunk 50: 0.45. Average BHR for chunk 200: 0.425.

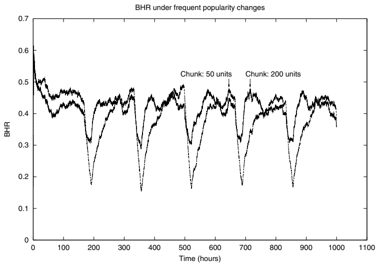


Fig. 15. Frequent popularity changes. Average BHR for chunk 50: 0.39. Average BHR for chunk 200: 0.41.

The effect of popularity changes. From the results presented so far, it is evident that the performance of the system depends not only on the chunk size but also on the frequency of popularity changes (and on how dramatic these changes are). Under a fixed popularity distribution, a small chunk ensures a better steady state BHR than a large chunk size. In practice the BHR under a small chunk size is never reached, as it requires a long adaptation period which is not available under frequent changes in popularity. This could lead to an overall performance that is worse than that achieved under a large chunk size. Figures 14 and 15 illustrate the BHR(24h) versus time for two different cases [8]. Fig. 14, corresponds to the case where demand changes occur rarely (only one at time instant 500). It is observed that in periods where popularity remains stable a smaller chunk provides for better BHR. For the periods that follow a popularity change the BHR reduces for both chunk sizes but the reduction is smaller for a large chunk, which quickly adapts to the new popularity (observe the shallow gap). A small chunk size, although performing better under static popularity, it is outperformed during periods of popularity changes as it needs a considerably larger time to catch-up with the new demand changes and consequently, the gaps are deeper. The average BHR over the entire observation window, is equal to 0.45 for the system that uses a chunk size of 50 units and 0.425 for the system that uses a chunk size of 200 units. On the other hand, in Fig. 15 where demand changes occur more often (every 170 hours), the system that uses a chunk size of 200 units achieves higher average BHR (0.41 over 0.39).

Adaptation to Changes of Popularity. To cope with sudden changes of popularity it is proposed that the system use a larger chunk size during periods when a change of popularity has just occurred. Sophisticated methods can detect the change of popularity by looking for sudden decreases of BHR and adjusting the chunk size automatically. In any case, once the cache content has been updated the system could switch back to a small chunk size in order to achieve higher ss-BHR.

[8] In order to achieve a fast convergence to the stready state, the initial state of the cache is considered to coincide with the allocation under HPF

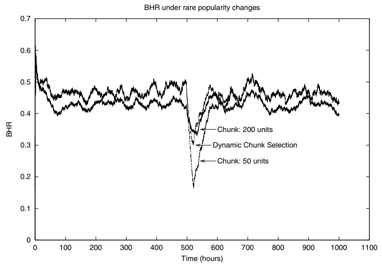


Fig. 16. Rare popularity changes & Dynamic Chunk Selection. Average BHR for chunk 50: 0.45. Average BHR for chunk 200: 0.425. Average BHR for dynamic chunk selection: 0.46.

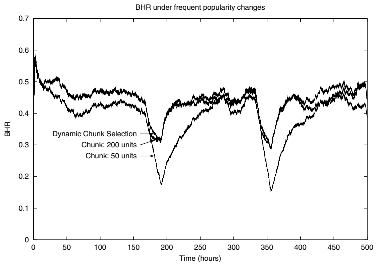


Fig. 17. Frequent popularity changes & Dynamic Chunk Selection. Average BHR for chunk 50: 0.39. Average BHR for chunk 200: 0.41. Average BHR for dynamic chunk selection: 0.44.

The benefits under a dynamic selection of the chunk size are illustrated in Fig. 16 and Fig. 17 where the BHR is depicted as a function of time. The parameters are the same as those in Fig. 14 and Fig. 15 respectively. From these figures it becomes clear that a change of the chunk size at the right moment combines the advantages of both chunk sizes and results in a better overall BHR performance than under the corresponding static schemes. Note that the average BHR under the dynamic chunk selection (0.46 (0.44) for the system of Fig. 16 (Fig. 17)) is higher than that under the fixed chunk size schemes for chunk size 50 (0.45 (0.39)) and chunk size 200 (0.425 (0.41)). The application of dynamic chunk selection corresponding to Fig. 14 is illustrated in Fig. 16.

6 Additional Considerations

The presented system has taken a rather simplistic approach, as far as the cost of the backbone bandwidth is concerned, by treating all video data as being equal. This has mainly been dictated by our desire not to obscure this first exposition of the effects of partial caching with additional parameters that are not directly related to this issue. In reality not all bits entail the same cost nor should be treated the same way. Some of the reasons for which it may be appropriate to differentiate among videos are:

Server-proxy distance: In general, the greater the distance between the server and the proxy (e.g. in number of hops), the higher the cost of fetching a video from that server. This mean that a cache hit results in a greater reduction of required bandwidth when the requested video belongs to a distant server.

Different link costs: Links may have different costs due to different bandwidth-availability/bandwidth-demand ratios. For example, the proxy could give preferential treatment to content that resides in an origin server that is situated behind a congested link. Loss and/or delay measurements could be used for the estimation of the congestion level.

Content differentiation: Some content could be given preferential treatment as requested by the content provider, e.g., some clips could be given some sort of priority in the cache, so as to be available upon a request of a popular web page that includes them, with a revenue collected for the special treatment.

All the cases of can be accommodated by using variable sized chunks. The presented system has used a variable sized chunk to cope with changes of popularity; chunks of different size could be used also within periods of static popularity to provide for differentiation as suggested in the aforementioned examples.

References

1. Scott A. Barnett, Gary J. Anido, and H.W. Beadle, "Caching policies in a distributed video on-demand system," in *Australian Telecommunication Networks and Applications Conference*, Sydney, Australia.
2. J.-P. Nussbaumer, B. V. Patel, F. Schaffa, and J. P. G. Sterbenz, "Networking requirements for interactive video on demand," *IEEE Journal on Selected Areas in Communications*, vol. 13,5, pp. 779–787, 1995.
3. Christos Papadimitriou, Srinivas Ramanathan, P Venkat Rangan, and Srihari Sampathkumar, "Multimedia information caching for personalized video-on-demand," *Computer Communications*, vol. 18, no. 3, Mar. 1995.
4. Subhabrata Sen, Jennifer Rexford, and Don Towsley, "Proxy prefix caching for multimedia streams," in *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, New York, Mar. 1999.
5. Zhi-Li Zhang, Yuewei Wang, D. H. C. Du, and Dongli Su, "Video staging: A proxy-server-based approach to end-to-end video delivery over wide-area networks," *IEEE/ACM Transactions on Networking*, vol. 8, no. 4, Aug. 2000.
6. Bing Wang, Subhabrata Sen, Micah Adler, and Don Towsley, "Proxy-based distribution of streaming video over unicast/multicast connections," Technical Report UMASS TR-2001-05, University of Massachusetts, Amherst, 2001.
7. S. Ramesh, I. Rhee, and K. Guo, "Multicast with cache (mcache): An adaptive zero-delay video-on-demand service," in *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, Anchorage, Alaska, Apr. 2001.
8. S.-H. Gary Chan and Fouad A. Tobagi, "Caching schemes for distributed video services," in *Proceedings of the IEEE International Conference on Communications (IEEE ICC)*, Vancouver, Canada, June 1999.
9. Markus Hofmann, T.S. Eugene Ng, Katherine Guo, Paul Sanjoy, and Hui Zhang, "Caching techniques for streaming multimedia over the internet," Tech. Rep., Bell Laboratories, May 1999.
10. Jussi Kangasharju, Felix Hartanto, Martin Reisslein, and Keith W. Ross, "Distributing layered encoded video through caches," in *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, Anchorage, Alaska, Apr. 2001.
11. Reza Rejaie, Haobo Yu, Mark Handley, and Deborah Estrin, "Multimedia proxy caching mechanism for quality adaptive streaming applications in the internet," in *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, Tel Aviv, Israel, Mar. 2000.
12. Philippe Flajolet, Gardy Danièle, and Thimonier Loys, "Birthday paradox, coupon collectors, caching algorithms and self-organizing search," *Discrete Applied Mathematics*, vol. 39, no. 3, pp. 207–229, 1992.
13. Predrag R. Jalenković, "Asymptotic approximation of the move-to-front search cost distribution and least-recently-used caching fault probabilities," *Annals of Applied Probability*, vol. 9, no. 2, pp. 430 – 464, 1999.

Utility Analysis of Simple FEC Schemes for VoIP

Parijat Dube[1] and Eitan Altman[1,2]

[1] INRIA, B.P.93, 06902 Sophia-Antipolis, France.{pdube,altman}@sophia.inria.fr
[2] C.E.S.I.M.O., Facultad de Ingeneria, Universidad de Los Andes, Mérida, Venezuela

Abstract. Forward Error Correction (FEC) is an attractive solution for recovering from packet losses in real-time applications. However, in many such applications, due to delay constraints, complex FEC schemes that may enable to approach the Shannon channel capacity cannot be used. In those cases, much simpler FEC schemes are implemented, in which a compressed copy of a packet is concatenated to some subsequent packet. In this paper we analyze the utility gained by such schemes in the case where losses are due to queueing congestion in the network. Loss probabilities in presence of FEC are obtained using sophisticated tools based on Ballot theorems. Our explicit expressions allows us to study numerically several approaches for adding FEC.

1 Introduction

Forward Error Correction (FEC) is considered to be an attractive solution for recovering from packet losses in real-time applications (over high speed networks [8,12]) without increasing latency. Though there are different proposals for FEC schemes,in this paper we shall focus on a simple FEC scheme that has been proposed [4] in the IETF (Internet Engineering Task Force) and implemented in audio tools like Freephone [1] and Rat [15]. The scheme consists in adding a redundant copy of the original packet to the tail of the subsequent packet. If a packet is lost in the network then it can be reconstructed from its redundant copy (if it is correctly received). It was suggested to use PCM coding for data in the original packet and a low bit rate coding (like LPC, GSM, or 2-bit ADPCM) for the redundant information. Proposals to enhance the performance of this simple FEC scheme include: (i) increasing the offset ϕ_0 between the original packet and its redundant copy [10,9] at the codec, (ii) adding multiple redundant copies of a packet in multiple subsequent ones [10], (iii) adding to a packet a redundant copy computed from a block of some preceding packets [5] etc.

Let the offset between the original packet and its redundant copy at the bottleneck be ϕ. In a recent work [6], treating ϕ as a parameter it has been shown via a queueing analysis that the simple FEC scheme *may* not lead to an improvement in audio quality. The assumptions made in [6] include: (i) a single bottleneck link with dedicated buffer for audio flows implementing FEC and (ii) a linear utility function (A *utility function* is an indication of the variation

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 226–239, 2002.

of the audio quality at the receiver as a function of the transmission rate). The authors showed that for a buffer size K, the ratio of the volume of the redundant information to the volume of the original information α, a Poisson arrival of packets, exponential distribution of service times and for ϕ less than the *scaled* buffer size (in terms of number of packets) $K_\alpha = \frac{K}{(1+\alpha)}$, adding FEC according to the simple FEC scheme always leads to a deterioration in quality caused by increased load in the network due to redundant data. They further showed that the same is also true for the limiting case $\phi \to \infty$, which forms an upper bound on the audio quality.

In their subsequent work [7], the authors showed that improvement in quality is possible with the simple FEC scheme when: (i) the (total) rate of flow(s) adding FEC is small compared to the total rate of the other flow(s), called *exogenous flows* not adding FEC and sharing the same bottleneck and (ii) with particular non-linear utility functions. It was also concluded that the audio quality is always an increasing function of the offset ϕ between the original packet and its copy. They also provided lower and upper bounds on the audio quality for case (i) with Poisson arrivals of packets but with exponential distribution of packet sizes for exogenous flows and deterministic packet sizes for audio flows implementing FEC.

It was argued in [6] that $\phi > K_\alpha$ may not be interesting as it could lead to unacceptable delays at the receiver. But it should be noticed that the actual offset between the original packet and its redundant copy at the bottleneck could become larger than the offset generated at the transmission codec due to multiplexing of different flows (either implementing FEC or not) at the bottleneck[1]. Moreover, in the case of large bandwidth, when many connections can be supported, a large offset does not generate necessarily a large delay, as the transmission time of a packet at the bottleneck is still very short.

In this paper we first complement the work in [6] and compute the performance for a fixed $\phi > K_\alpha$. It turns out that this case requires much more sophisticated computations than those in [6]. Using these results, we then compute the performances for the case of random offset obtained in situations where several flows are multiplexed. In particular, we obtain expressions for the expected audio quality for general utility functions. We use these results for an extensive numerical investigation using six different utility functions.

In this paper we deal with a Markovian framework: Poisson arrivals of packets and exponential distribution of all packet lengths and with the simple FEC scheme with $\phi_0 = 1$. We derive expressions for the quality function of an audio flow implementing FEC, multiplexed with: (i) other audio flows implementing FEC and/or (ii) exogenous flows not implementing FEC and different utility functions and Φ varying from 1 to ∞. These calculations will finally help us in the evaluation of the expected value of the quality function and the (possible) gain achievable with FEC in a more realistic framework of multiplexing of different flows.

[1] In presence of other multiplexed traffic, the offset ϕ is not deterministic anymore: it becomes a random variable Φ, taking values from ϕ_0 (the offset generated at the codec) to ∞.

2 Model

In this section we shall use a simple (same as in [6]) $M/M/1/K$ queue to model the network and thus the loss process of audio packets. In other words, we assume that packets arrive at the bottleneck [2] according to a Poisson process with intensity λ and the lengths of the packets are exponentially distributed with parameter μ. Let $\rho = \lambda/\mu$, be the intensity of the audio traffic and with $\rho < 1$, the loss probability of a packet in steady state is given by (see [11]), $\pi_{\rho,K} = \frac{1-\rho}{1-\rho^{K+1}}\rho^K$, and for $\rho = 1$ it is equal to, $\pi_{\rho,K} = \frac{1}{K+1}$.

Let $U(\alpha)$ be the utility function associated with receiving only a compressed version of a packet with a redundancy factor α, $0 \leq \alpha \leq 1$. We assume that $U(\alpha)$ is non-decreasing in α and $lim_{\alpha\to 0}U(\alpha) = 0$. Let Y_n [3] be a random variable defined as,

$$Y_n = \begin{cases} 0 & \text{if packet } n \text{ is lost} \\ 1 & \text{if packet } n \text{ is correctly received} \end{cases}$$

2.1 Case I: All Flows Implementing FEC

We consider the case when an audio flow implementing FEC (we shall refer to this flow as the *tagged* flow) shares the bottleneck with other audio flows also implementing same type of FEC and arriving as independent Poisson processes. Let λ_a be the total arrival rate of all other audio flows at the bottleneck. For each flow, the redundant information of a packet is located in the next. Thus $j+1$ packet of an audio flow will contain in addition to its own useful information some redundant information of the jth packet of the same flow. Let $\bar{\rho} = (\lambda + \lambda_a)/\mu$. We shall consider two scenarios concerning the influence of redundancy on the packet size.

- *Constant packet size model*: We assume that the size of a packet is not affected by adding redundancy and thus redundancy is a overhead. The more the redundancy the less is the *useful* information carried by the packet.
- *Constant amount of useful information*: We assume that the packet size increases when adding redundancy (and thus increases with α) so that the amount of useful information contained in a packet is unchanged. Increasing packet size has an impact both on the service times, and on the number of packets that can be stored at buffers. To account for this we assume [6]:
  - the service time is exponentially distributed with parameter $\mu_\alpha = \frac{\mu}{1+\alpha}$. Thus $\bar{\rho}_\alpha = \bar{\rho}(1+\alpha)$.
  - The amount of buffering is diminished by $1+\alpha$. Thus the scaled buffer size is $K_\alpha = \lfloor \frac{K}{1+\alpha} \rfloor$

[2] Most of the losses from a flow occur in the router having the smallest available bandwidth in the chain of routers, so that one may model the whole chain of routers by one single router called the *bottleneck*. This assumption has both theoretical and experimental [14,2] justification.

[3] subscript n is for the nth packet of the total flow at the queue

The loss probability in the presence of redundancy for this case is $\pi_{\bar{\rho}_\alpha,K_\alpha} = \frac{1-\bar{\rho}_\alpha}{1-(\bar{\rho}_\alpha)^{K_\alpha+1}}(\bar{\rho}_\alpha)^{K_\alpha}$.

In the following analysis we shall denote by $P_\alpha(.)$, the probabilities evaluated at μ_α and K_α. Observe that by taking $\alpha = 0$ in $P_\alpha(.)$ and $\pi_{\bar{\rho}_\alpha,K_\alpha}$ we get the corresponding expressions for the case of constant packet size as $K_0 = K$ and $\bar{\rho}_0 = \bar{\rho}$. Let $P_\alpha(Y_{n+\Phi} = 1, \Phi = \phi | Y_n = 0)$ be the probability that due to multiplexing there are $\phi - 1$ packets between two consecutive packets of the tagged audio flow at the bottleneck and that the $n+\phi$th packet is received correctly given that the nth packet is lost. Without loss of generality we assume that packet n belongs to the tagged audio flow. We define below the expected quality function $Q(\alpha)$ for the two cases.

- Constant packet size

$$Q(\alpha) = \tag{1}$$

$$P_0(Y_n = 1)U(1-\alpha) + U(\alpha)P_0(Y_n = 0)\sum_{\phi=1}^{\infty} P_0(Y_{n+\Phi} = 1, \Phi = \phi | Y_n = 0)$$

$$= U(1-\alpha) - \pi_{\bar{\rho},K}(U(1-\alpha) - U(\alpha)\sum_{\phi=1}^{\infty} P_0(Y_{n+\Phi} = 1, \Phi = \phi | Y_n = 0)).$$

- Constant amount of useful information: as the amount of useful information carried by a packet remains unchanged we define

$$Q(\alpha) = \tag{2}$$

$$P_\alpha(Y_n = 1)U(1) + U(\alpha)P_\alpha(Y_n = 0)\sum_{\phi=1}^{\infty} P_\alpha(Y_{n+\Phi} = 1, \Phi = \phi | Y_n = 0)$$

$$= U(1) - \pi_{\bar{\rho}_\alpha,K_\alpha}(U(1) - U(\alpha)\sum_{\phi=1}^{\infty} P_\alpha(Y_{n+\Phi} = 1, \Phi = \phi | Y_n = 0)).$$

We will next find the complementary probability $P_\alpha(Y_{n+\Phi} = 0, \Phi = \phi | Y_n = 0)$. In steady state we need to evaluate the probability $P_\alpha(Y_\Phi = 0, \Phi = \phi | Y_0 = 0)$ which in terms of the buffer level is $P_\alpha(X_\Phi = K_\alpha, \Phi = \phi | X_0 = K_\alpha)$, where X_k is the queue size as seen by an arriving kth packet. We will now evaluate $P_\alpha(X_\Phi = K_\alpha, \Phi = \phi | X_0 = K_\alpha)$ for $\phi = \{1, 2, \ldots\}$ using results from ballot theorem.

Let A_0 be the event of the loss of the 0th packet and A_ϕ the event of the loss of the ϕth packet. We consider the following cases:

- Case 1: no packet is lost in the interval between the happening of A_0 and $A_\phi \Rightarrow$ the first lost packet after the 0th packet is the ϕth packet.
- Case 2: $i(1 \leq i \leq \phi - 1)$ packets are lost between A_0 and A_ϕ.

We consider the two cases separately. The main difficulty in the case when the redundancy-offset is larger than the (scaled) buffer, i.e., $\phi > K_\alpha$ is that paths with this property may cause the buffer to become empty (even several times). This makes the analysis substantially more complex than in [6].

To proceed we first consider a general path that starts when there are K_α packets in the buffer, ends with K_α packets in the buffer, contains η events (arrivals of Poisson process with parameter $\lambda+\lambda_a$ and departures) and no packets are lost. Let ζ_η be the probability of such a path. We will now express $P_\alpha(X_\Phi = K_\alpha, \Phi = \phi | X_0 = K_\alpha)$ for the two cases in terms of this probability.

- Case 1: for the ϕ packet to be the first lost packet there must be an epoch after the arrival of the $\phi - 1$ packet that the buffer is full and the next event is an arrival. Upto that epoch the number of departures must be equal to the number of arrivals. The number of arrivals upto that epoch is $\phi - 1$ and so the number of departures is $\phi - 1$ to yield a total number of $\eta = 2(\phi - 1)$ events. Thus the probability for this case is: $\zeta_{2(\phi-1)} \frac{\lambda}{\lambda+\lambda_a+\mu_\alpha}$. However we also require that all the $\phi - 1$ arrivals should be from other audio flows (coming as a Poisson process with intensity λ_a). Hence the joint probability that there are $\phi - 1$ packets from other flows (due to multiplexing) at the bottleneck and the ϕth packet is lost is $\zeta_{2(\phi-1)} \frac{\lambda}{\lambda+\lambda_a+\mu_\alpha} \left(\frac{\lambda_a}{\lambda_a+\lambda}\right)^{\phi-1}$.
- Case 2: First we look at $i = 1$. Let the lost packet be the jth arrival $(1 \leq j \leq \phi - 1)$. Thus there should be no loss from the instant of the loss of 0th packet till the epoch before the arrival of the jth packet. Since the queue size is K_α at A_0 and since the jth packet is the first to be lost, there must be an epoch after the arrival of the $j - 1$ packet that the buffer is again full and the next event is an arrival. Again, upto this epoch, the number of departures must be equal to the number of departures. The number of arrivals upto that epoch is $j - 1$, so the number of departures is $j - 1$ yielding a total number of $\eta = 2(j - 1)$. Thus the probability of the path is: $\zeta_{2(j-1)} \frac{\lambda_a}{\lambda+\lambda_a+\mu_\alpha} \zeta_{2(\phi-j-1)} \frac{\lambda}{\lambda_a+\lambda+\mu_\alpha}$. Also, since we require that all the $\phi - 1$ arrivals should be from other audio flows we have the probability of the path as $\zeta_{2(j-1)} \frac{\lambda_a}{\lambda+\lambda_a+\mu_\alpha} \zeta_{2(\phi-j-1)} \frac{\lambda}{\lambda_a+\lambda+\mu_\alpha} \left(\frac{\lambda_a}{\lambda+\lambda_a}\right)^{\phi-2}$
  Since j can take values from 1 to $\phi - 1$, the total probability for this case is $\left(\frac{\lambda_a}{\lambda+\lambda_a}\right)^{\phi-2} \sum_{j=1}^{\phi-1} \zeta_{2(j-1)} \frac{\lambda_a}{\lambda+\lambda_a+\mu_\alpha} \zeta_{2(\phi-j-1)} \frac{\lambda}{\lambda+\lambda_a+\mu_\alpha}$.
  Now consider the case when $i = 2$. We will first calculate the probability of a single path with two losses before the loss of the ϕ packet. Let the first loss be of j_1 and second of j_2 packet $(j_1 < j_2, 1 \leq j_1 \leq \phi - 2, 2 \leq j_2 \leq \phi - 1)$. By previous arguments the probability of such a path is: $\zeta_{2(j_1-1)} \zeta_{2(j_2-j_1-1)} \left(\frac{\lambda_a}{\lambda+\lambda_a+\mu_\alpha}\right)^2 \zeta_{2(\phi-j_2-1)} \frac{\lambda}{\lambda+\lambda_a+\mu_\alpha}$. The total probability for this case is (taking into consideration the requirement that all the $\phi - 1$ packets should be from other audio flows):

$$\left(\frac{\lambda_a}{\lambda+\lambda_a}\right)^{\phi-3} \sum_{j_1=1}^{\phi-2} \sum_{j_2=j_1+1}^{\phi-1} \zeta_{2(j_1-1)} \zeta_{2(j_2-j_1-1)} \zeta_{2(\phi-j_2-1)} \frac{\lambda \lambda_a}{(\lambda+\lambda_a+\mu_\alpha)^3}.$$

In general for $i = k(1 \leq k \leq \phi - 1)$, we can write the probability as:

$$\left(\frac{\lambda_a}{\lambda + \lambda_a}\right)^{\phi-(k+1)} \sum_{j_1=1}^{\phi-k} \sum_{j_2=j_1+1}^{\phi-k+1} \cdots \sum_{j_k=j_{k-1}+1}^{\phi-1} \zeta_{2(j_1-1)} \zeta_{2(j_2-j_1-1)} \cdots$$

$$\cdots \zeta_{2(j_k-j_{k-1}-1)} \zeta_{2(\phi-j_k-1)} \left(\frac{\lambda_a}{\lambda + \lambda_a + \mu_\alpha}\right)^k \frac{\lambda}{\lambda + \lambda_a + \mu_\alpha}.$$

Thus to complete the analysis we need to evaluate the quantity ζ_η where $\eta = 2, 4, 6, \ldots$. To evaluate this we use the result of [13] which is based on ballot theorems. In this paper the authors have obtained the following general expression for $\zeta_\eta(\iota, v)$, i.e., the probability of a path that starts with $\iota(1 \leq \iota \leq K_\alpha)$ packets in the buffer ends with $v(1 \leq v \leq K_\alpha)$ packets in the buffer and contains $\eta(0 \leq \eta \leq 2\phi + \iota - v)$ events (arrivals and departures) and no packets are lost:

$$\zeta_\eta(\iota, v) = \xi_\eta(\iota, v) + \sum_{r=1}^{\mathcal{R}} W_\iota Y^{r-1} [Z(\iota, v)],$$

where $\xi_\eta(\iota, v)$ is the probability of a path that starts with ι packets in the buffer ends with v packets in the buffer and contains η events ($\mid \iota - v \mid \leq \eta \leq 2n + \iota - v$, $1 \leq \iota \leq K_\alpha$, $1 \leq v \leq K_\alpha$) and the buffer never empties along this path, $\mathcal{R} = 1 + \frac{\eta - \iota - v}{2}$, W_ι and Z are $\mathcal{R}$-length row and Y is an $\mathcal{R}X\mathcal{R}$ matrix defined as

$$W_\iota = \left(\xi_\iota(\iota, 0), \xi_{\iota+2}(\iota, 0), \ldots, \xi_{\iota+2(\mathcal{R}-1)}(\iota, 0)\right),$$

$$Z(\iota, v) = \left(\xi_{\eta-\iota}(0, v), \xi_{\eta-\iota-2}(0, v), \ldots, \xi_{\eta-\iota-2(\mathcal{R}-1)}(0, v)\right),$$

$$Y = \begin{pmatrix} 0 & \xi_2(0,0) & \xi_4(0,0) & \ldots & \xi_{2(\mathcal{R}-1)}(0,0) & \\ 0 & 0 & \xi_2(0,0) & \ldots & \xi_{2(\mathcal{R}-2)}(0,0) & \\ \vdots & \vdots & \vdots & \vdots & \vdots & \\ \vdots & \vdots & \vdots & \vdots & \vdots & . \\ \vdots & \vdots & \vdots & \vdots & \vdots & \\ 0 & 0 & 0 & \ldots & \xi_2(0,0) & \\ 0 & 0 & 0 & \ldots & 0 & \end{pmatrix}.$$

For our analysis we need $\zeta_\eta (= \zeta_\eta(K_\alpha, K_\alpha))$. Thus we get,

$$\zeta_\eta(K_\alpha, K_\alpha) = \zeta_\eta = \xi_\eta(K_\alpha, K_\alpha) + \sum_{r=1}^{\mathcal{R}} W_{K_\alpha} Y^{r-1} Z(K_\alpha, K_\alpha),$$

where $\mathcal{R} = 1 + \frac{\eta - 2K_\alpha}{2} = 1 + \frac{\eta}{2} - K_\alpha$. The steps for evaluation of $\xi_\eta(\iota, v)$ are available in [13]. Thus we can write $P_\alpha(X_\Phi = K_\alpha, \Phi = \phi | X_0 = K_\alpha)$ as

$$P_\alpha(X_\Phi = K_\alpha, \Phi = \phi | X_0 = K_\alpha) = \tag{3}$$

$$\zeta_{2(\phi-1)} \frac{\lambda}{\lambda_a + \lambda + \mu_\alpha} \left(\frac{\lambda_a}{\lambda + \lambda_a}\right)^{\phi-1} + \frac{\lambda}{\lambda_a + \lambda + \mu_\alpha}$$

$$\sum_{k=1}^{k=\phi-1}\left[\sum_{j_1=1}^{\phi-k}\sum_{j_2=j_1+1}^{\phi-k+1}\cdots\cdots\sum_{j_k=j_{k-1}+1}^{\phi-1}\zeta_{2(j_1-1)}\zeta_{2(j_2-j_1-1)}\cdots\right.$$
$$\left.\cdots\zeta_{2(j_k-j_{k-1}-1)}\zeta_{2(\phi-j_k-1)}\left(\frac{\lambda_a}{\lambda_a+\lambda+\mu_\alpha}\right)^k\left(\frac{\lambda_a}{\lambda+\lambda_a}\right)^{\phi-(k+1)}\right].$$

Thus knowing $P_\alpha(X_\Phi = K_\alpha, \Phi = \phi|X_0 = K_\alpha)$ we have $P_\alpha(Y_\Phi = 0, \Phi = \phi|Y_0 = 0)$ and thus we have $P_\alpha(Y_\Phi = 1, \Phi = \phi|Y_0 = 0) = P_\alpha(\Phi = \phi|Y_n = 0) - P_\alpha(Y_\Phi = 0, \Phi = \phi|Y_0 = 0) = P_\alpha(\Phi = \phi) - P_\alpha(Y_\Phi = 0, \Phi = \phi|Y_0 = 0,\ 1 \leq \phi < \infty$. We next find the distribution of Φ. Observe that $\Phi - 1$ is the number of packets of other flows (other audio flows arriving as an independent Poisson process with parameter λ_a). Thus Φ is geometrically distributed: $P_\alpha(\Phi = n) = \gamma^{n-1}(1-\gamma)$ for $n \geq 1$, with $\gamma = \gamma_1 = \frac{\lambda_a}{\lambda+\lambda_a}$. Knowing this we can evaluate the expected quality function $Q(\alpha)$ using (1),(2).

2.2 Case II: Multiplexing between Audio Flows Implementing FEC and Exogenous Flows for the Constant Packet Size Case

The case when an audio flow implementing FEC shares the bottleneck with other audio flows also implementing the same FEC scheme and with some other exogenous flows not implementing FEC can be analysed similarly for the case of constant packet size. We model the packets of exogenous arriving as an independent (from audio flows) Poisson process with intensity λ_e and hence the total arrival process is again a Poisson process with parameter $\lambda + \lambda_a + \lambda_e$. Further we assume that the distribution of the size of all packets of all flows is exponential with parameter μ. With $\bar{\rho} = \frac{\lambda+\lambda_a+\lambda_e}{\mu}$, the loss probability of a packet is steady state (with $\bar{\rho} < 1$) is $\pi_{\bar{\rho}} = \frac{1-\bar{\rho}}{1-\bar{\rho}^{K+1}}\bar{\rho}^K$, and for $\bar{\rho} = 1$ it is equal to $\pi_{\bar{\rho}} = \frac{1}{(1+K)}$. The expressions for expected quality can be obtained from the expressions for the case without exogenous but with λ_a replaced by $\lambda_a + \lambda_e$everywhere. For the case of constant amount of useful information it is not clear how we should scale the service rate and the buffer size for this scenario.

3 Numerical Examples with Different Utility Functions

We will next evaluate the expected quality achieved using different utility functions. The utility functions that we consider are used in [6,7] and are of the form proposed in [10,16]. We define: $U_1(\alpha) = \alpha$, $U_2(\alpha) = \sqrt{\alpha}$, $U_3(\alpha) = \alpha^{1/10}$, $U_4(\alpha) = esc(\alpha - \alpha_0)\left(\frac{1-\cos(\pi\alpha)}{2}\right)^{1/10}$, $U_5(\alpha) = \left(\frac{1-\cos(\pi\alpha)}{2}\right)^{1/3}$ and $U_6(\alpha) = 1$ if $\alpha > 0$ otherwise $U_6(\alpha) = 0$. All utilities are concave starting at some minimum and reach $U(\alpha) = 1$ for $\alpha = 1$. The least concave is the linear utility function U_1, which is thus proportional to the amount of information which is well received. The most concave function is U_6. Although it does not represent a utility of any real application, it can be used to obtain an upper bound on the gain achieved with employing FEC. Indeed, we see that U_6 is larger than any other utility, so it follows that in both models described by (1) and (2), it should give the best quality. The utility U_4 is zero for $\alpha \leq \alpha_0$. This is typical for real time applications with a minimum hard constraint. For example, the constraint may represent the fact that the throughput of existing codecs for voice applications cannot go beneath some bound. We shall plot the quality (expected) function with these six utility functions under different scenarios. For the plots we denote by M the actual buffer size and by ρ that total load. Also note that **in all the plots that follow the utility functions are represented as:** $* - U_1$, $+ - U_2$, $. - U_3$, $o - U_4$, $- \ U_5$ **and finally** $- - U_6$.

1. *A single audio flow implementing FEC traversing the bottleneck (no multiplexing):* When a single audio flow traverses the bottleneck (i.e., $\lambda_a = \lambda_e = 0$), then the spacing between a packet and its redundancy at the bottleneck is the same as that generated at the transmission codec. We take $\lambda = 1$ and $\rho = \lambda/\mu$. We next plot the quality function with the six utility functions for this scenario and for the cases when: (i) the packet sizes (and hence the buffer sizes) remains unchanged after adding redundancy and the quality function is given by (1) in Figs. 1, 2 (ii) when the packet sizes changes after adding redundancy and the quality function is given by (2) in Figs. 3, 4.

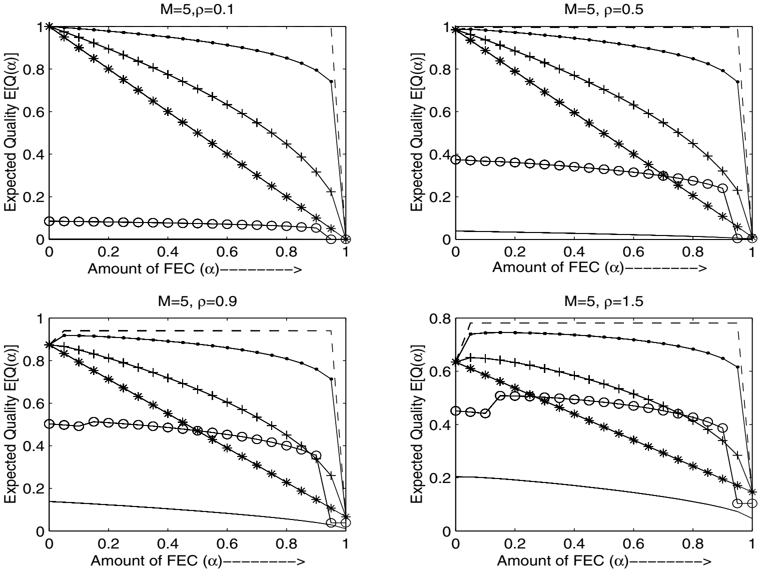


Fig. 1. A single audio flow implementing simple FEC traversing the bottleneck with quality function given by (1).

We observe that

– as expected, U_6 gives an upper bound for the quality. More generally, for two utility functions, U_j and U_i, if $U_i \geq U_j$ for all α, then the corre-

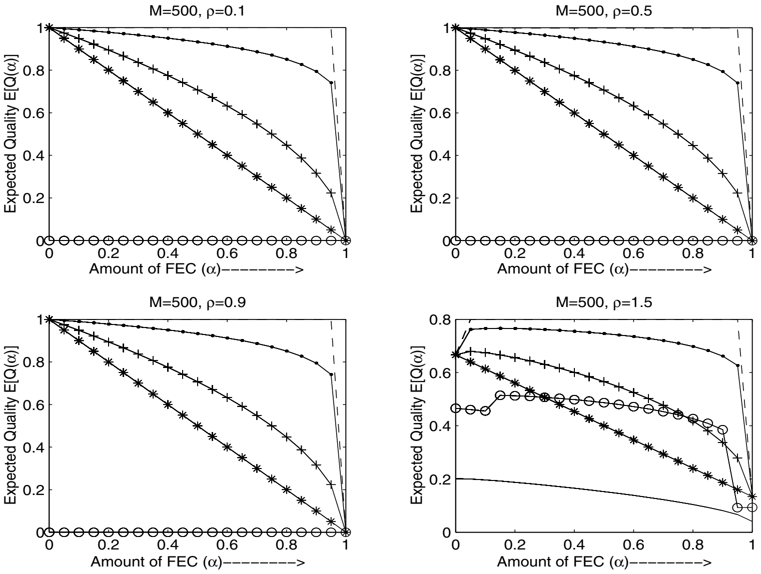


Fig. 2. A single audio flow implementing simple FEC traversing the bottleneck with quality function given by (1).

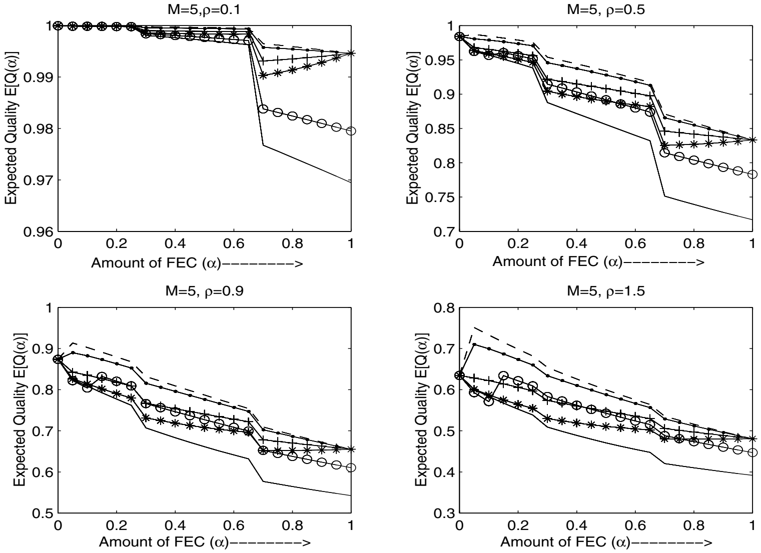


Fig. 3. A single audio flow implementing simple FEC traversing the bottleneck with quality function given by (2).

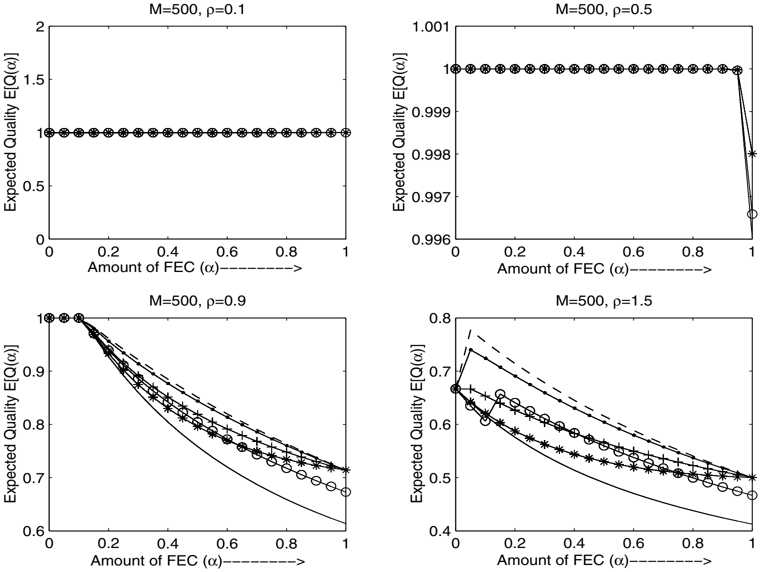


Fig. 4. A single audio flow implementing simple FEC traversing the bottleneck with quality function given by (2).

sponding quality is larger for both cases (1) and (2). This is confirmed in the figures, taking into account that $U_6 \geq U_3 \geq U_2 \geq U_1$.

- For large buffer size ($M = 500$) the quality is almost the same for all $\rho \leq 0.9$; the reason is that the loss probabilities are very small and almost all contribution for the quality is from the utility of unlost packets.
- For $\rho > 1$ we see that we gain by adding FEC (for any buffer size, and for both cases described by (1) and (2)), for utilities U_3, U_4, U_6. For small buffer size $M = 5$ we gain also for $\rho = 0.9$ when using utility U_3, U_4, U_6 for both the cases but for very low α, 0.1 or less.
- The linear utility function always decreases with FEC for any ρ and any M.
- The quality is higher with constant information model than with constant packet size model.

2. *An audio flow implementing simple FEC sharing the bottleneck with other audio flows implementing the same FEC:* This is the case I (Sec. 2.1) of our analysis. We take $\lambda = 0.1$ and $\lambda_a = 0.9$ and $\rho = (\lambda + \lambda_a)/\mu$. We plot the *expected* quality function with the six utility functions for this scenario and for the cases when: (i) the packet sizes (and hence the buffer sizes) remains unchanged after adding redundancy and the quality function is given by (1) in Figs. 5, 6 (ii) when the packet sizes changes after adding redundancy and the quality function is given by 7, 8. Due to multiplexing Φ takes values in $\{1, 2, \ldots\}$. For our numerical calculations we restricted to $\phi = \{1, 2, \ldots, 8\}$ as for $\phi \geq 9$, the contribution to $Q(\alpha)$ was negligible. Thus for buffer size of 5 we will have the spacing exceeding the buffer size. Observe that the

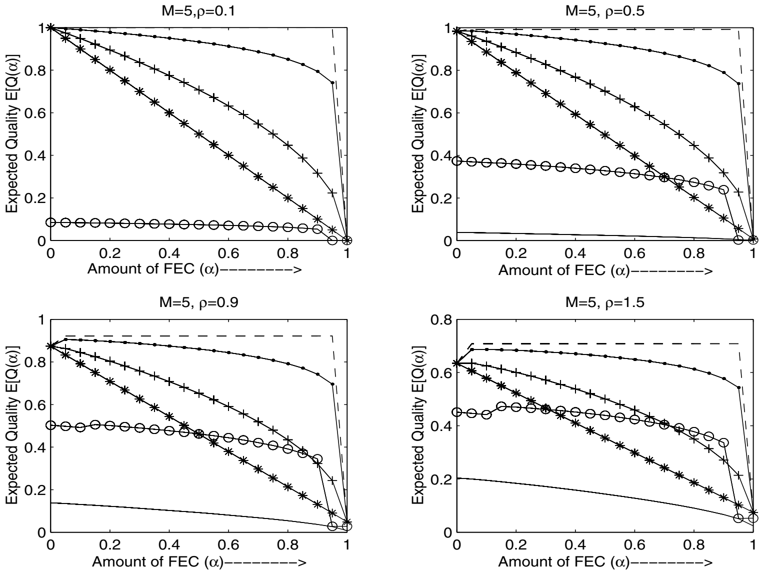


Fig. 5. An audio flow implementing simple FEC sharing the bottleneck with other audio flows implementing the same FEC scheme with quality function given by (1).

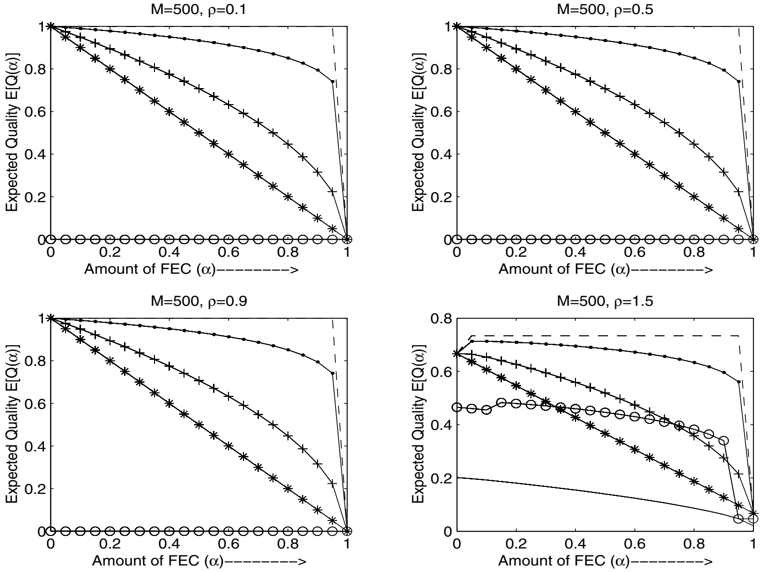


Fig. 6. An audio flow implementing simple FEC sharing the bottleneck with other audio flows implementing the same FEC scheme with quality function given by (1).

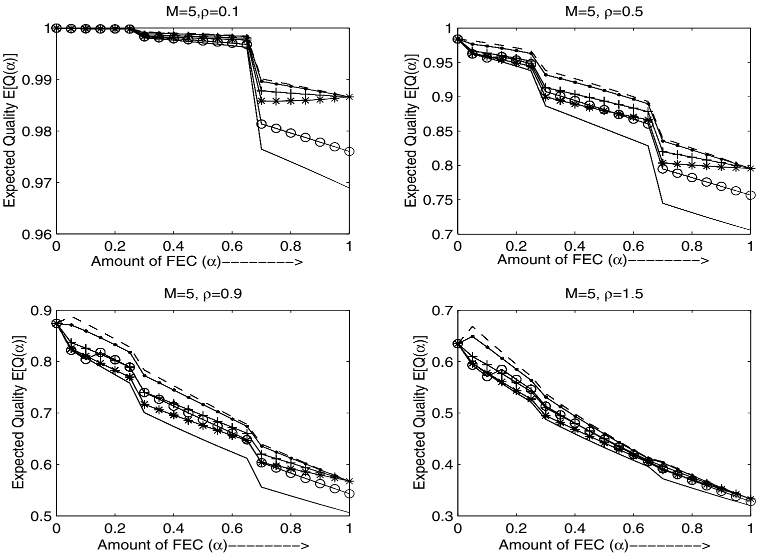


Fig. 7. An audio flow implementing simple FEC sharing the bottleneck with other audio flows implementing the same FEC scheme with quality function given by (2).

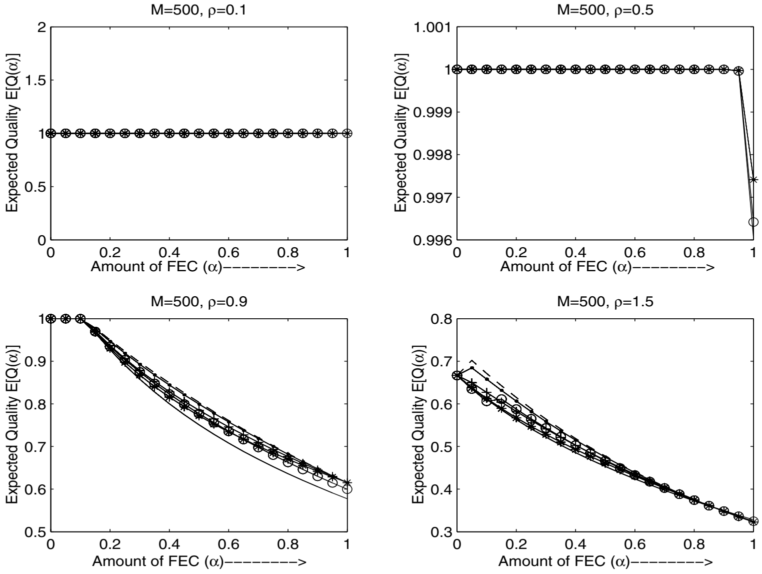


Fig. 8. An audio flow implementing simple FEC sharing the bottleneck with other audio flows implementing the same FEC scheme with quality function given by (2).

plots (5) and (6) are almost similar to plots (1) and (2) respectively. From (7) and (8) We observe that for large $\rho(>1)$, the buffer size does not affect the expected quality.

Also from all the plots it is observed that U_5 always gives a lower bound on expected quality for large $\alpha(\geq 0.2)$.

4 Conclusion

In this paper we studied the (possible) gain obtained with a simple FEC scheme when the losses are due to buffer overflow at the bottleneck. We obtained the loss probabilities in the presence of FEC using ballot theorem. To this end we generalize the analysis in [6] (which was for $\phi < K_\alpha$) and computed the loss probability for a fixed ϕ taking values from 0 to ∞. Using these results we obtained the expressions for the expected audio quality for general utility functions and then utilised the tools developed for a detailed numerical studies with six utility functions under various scenarios of multiplexing at the bottleneck. Our future work is to analyse delay aware utility functions [3] and to do utility analysis of other more intelligent and efficient FEC schemes to quantize the (possible) gain.

References

1. A. V. Garcia, S. Fosse-Parisis. *(FreePhone Audio Tool) High-Speed Networking Group.* INRIA, Sophia Antipolis, France.
2. J. C. Bolot. End-to-End Delay and Loss Behavior in the Internet. *Proc. Sigcomm '93*, pages 289–298, 1993.
3. C. Boutremans, J. L. Boudec. Adative Delay Aware Error Control for Internet Telephony. *Proc. IPTEL 2001*, 2001.
4. C. Perkins, L. Kouvelas, O. Hodson, V. Hardman. *RTP payload for redundant audio data.* RFC 2198 (1997).
5. D. R. Figueiredo, E. de Souza e Silva. Efficient Mechanisms for Recovering Voice Packets in the Internet. *Proc. IEEE Globecom '99*, 1999.
6. E. Altman, C. Barakat and V. M. Ramos R. Queueing Analysis of Simple FEC Schemes for IP Telephony. *Proc. IEEE Infocom 2001*, April 2001.
7. E. Altman, C. Barakat, V. M. Ramos R. *On the Utility of FEC Mechanisms for Audio Applications.* Proc. Second International Workshop on Quality of Future Internet Services, Qofis 2001, 24-26 Sept., 2001, Coimbra, Portugal. See also INRIA Research Report No. RR-3998 at http://www-sop.inria.fr/mistral/personnel/Eitan.Altman/perf.html.
8. E. W. Biersack. Performance Evaluation of FEC in ATM Networks. *Proc. ACM Sigcomm '92*, pages 248–257, Aug. 1992.
9. I. Kouvelas, O. Hodson, V. Hardman, J. Crowcroft. Redundancy Control in Real-Time Internet Audio Conferencing. *Proc. of AVSPN '97, Aberdeen, Scotland*, Sept. 1997.
10. J. C. Bolot, S. Fosse-Parisis, D. Towsley. Adaptive FEC-Based Error Control for Interactive Audio in the Internet. *Proc. IEEE Infocom 1999.*
11. L. Kleinrock. *Queueing Systems, Vol. I.* John Wiley, New York, 1976.

12. N. Shacham, P. McKenney. Packet Recovery in High-Speed Networks Using Coding and Buffer MAnagement. *Proc. IEEE Infocom '90*, pages 124–131, May 1990.
13. O. Gurewitz, M. Sidi, I. Cidon. The Ballot Theorem Strikes Again: Packet Loss Process Distribution. *IEEE Trans. on Information Theory*, 46(7):2588–2595, 2000.
14. O. J. Boxma. *Sojourn Times in Cyclic Queues: the influence of the slowest server, Computer Performnace and Reliability.* Elsevier Science Pubs. B. V. (North-Holland), 1988.
15. Mice Project. *(RAT: Robust Audio Tool) Multimedia Integrated Conferencing for European Researchers.* University College London, U.K.
16. S. Shenker. Fundamental Design Issues for the Future Internet. *IEEE Journal on Selected Areas in Communication*, 13(7), September 1995.

A Power Saving Architecture for Web Access from Mobile Computers

Giuseppe Anastasi[1], Marco Conti[2], Enrico Gregori[2], and Andrea Passarella[1]

[1]University of Pisa, Dept. of Information Engineering
Via Diotisalvi 2 - 56126 Pisa, Italy
{g.anastasi, a.passarella}@iet.unipi.it,

[2]CNR - CNUCE Institute
Via G. Moruzzi, 1 56124 PISA—Italy
{marco.conti, enrico.gregori}@cnuce.cnr.it,

Abstract. This work proposes new power-saving strategies for mobile access to the Web. User mobility is a key factor in the evolution of Web services. Unfortunately, the legacy approach for Web access is very inefficient when applied to mobile users. One of the critical issues is the inefficient usage of energetic resources when adopting the legacy TCP/IP architecture for Web access from mobile devices. In this paper we address this problem by proposing a new architecture, namely *PS-Web*, which works at the transport layer and exploits some knowledge about the application behavior. PS-Web is transparent with respect to the application and independent from the sub-network technology. We implemented a prototype of PS-Web. Experimental results provided by this prototype have shown that a relevant energy savings (about 70% on average) can be achieved with respect to the legacy TCP/IP approach. Furthermore, power saving is obtained without a significant degradation in the QoS perceived by the users. Specifically, PS-Web introduces almost neglible additional delays (with respect to the legacy approach) in the downloading of a Web page.

1 Introduction

The *Mobile Internet* is emerging as one of the most promising fields in the area of computer networking. The Internet explosion in the last years has demonstrated that accessing information of some interest in the same moment they are needed is a valuable opportunity. In this context, the concept of *mobility* adds a new dimension: information is carried directly to the user at *any time* and *any place*. However, integrating mobile computers in the legacy Internet scenario is still a challenging problem for a number of reasons. Internet protocols and applications were designed with the implicit assumption that links are wired and hosts do not change their position in time. Mobile computers have less computation and storage resources with respect to desktop computers. Furthermore, they usually connects through wireless links that are characterized by lower bandwidth and greater bit error rate with respect to wired links. Finally, mobile computers have a limited energy autonomy since they are battery-fed. Hence, the use of legacy solutions causes a non-optimal usage of the

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 240-251, 2002.

system that heavily limits the growth of the mobile Internet. In particular, the scarcity of energy resources is a very limiting factor [7, 11, 15].

In principle, energy-related problems could be solved by either increasing the battery capacity or reducing the energy consumption. Projections on progresses in battery technology show that only small improvements in the battery capacity are expected in next future [20]. Hence, it is vital to manage energy efficiently.

Strategies for energy saving have been investigated at several layers including the physical-layer transmissions, the operating system, the network protocols and the application level [8,10,14,16,18,19,22,23,24]. In this paper we focus on strategies aiming at reducing the energy consumed due to networking. Although this component only accounts for about 10% of the total consumption in current notebooks [13], it increases to approximately 50% in hand-held computers (palm top, PDA, etc.) [12]. Hence, it becomes very important to design a power efficient networking subsystem.

Based on experimental measurements, [21] and [13] conclude that the only way to actually reduce the networking component of the energy consumption consists in switching the wireless network interface off during inactivity periods. Works in [12] and [13] show that the legacy TCP/IP network architecture may have a negative impact on the energy consumption and propose to exploit the *Indirect-TCP* approach [2]. A further improvement could be achieved by exploiting some knowledge about the application behavior. According to this evidence, power management should be controlled at higher layers, potentially even at the application layer [12,13,21].

In this paper we propose new energy-saving strategies implemented in software network protocols. Specifically, we operate at the transport and application layers and use an application-dependent approach in the sense that envisaged strategies exploit some characteristics of the application. However, the proposed solutions do not require any modification to the application itself.

We focus on Web services but our design is *modular* and could be easily adapted to any other network application. Web choice is justified by several reasons. First, it is today the most widely used Internet application and is seriously candidate to become the killer application for the mobile Internet too. Furthermore, Web users are typically sensitive to delays. Hence, achieving a significant reduction in the energy consumption while maintaining an acceptable Quality of Service (QoS) level is a very challenging task.

We defined a new architecture, throughout referred to as *PS-Web* (Power Saving Web), which allows mobile users to exploit Internet Web services with a QoS similar to the one provided by the legacy network architecture based on the TCP/IP protocol stack, but with a significant reduction in the energy consumption. The PS-Web architecture is based on the Indirect-TCP model [2], i.e., the TCP connection between the browser and the Web server is split into two connections: one between the browser (on the mobile computer) and an Access Point (at the border between the wireless and wired networks), and the other one between the Access Point and the Web server. Unlike the solution proposed in [12], however, a simplified transport protocol is used between the mobile host and the Access Point. Furthermore, inactivity timeouts and sleeping times used to switch off and on the network interface are not fixed – as in [21] and [12] – but are adjusted dynamically based both on information about the past history collected on-line and on statistical models of Web traffic pattern available in the literature. The Access Point works as a Power Saving Proxy Web, i.e., a Proxy Web with power saving support for mobile users. Specifically, it implements a pre-fetching mechanism.

Experimental results obtained on a prototype implementation of the PS-Web architecture based on a IEEE 802.11 WLAN [9] have shown that the PS-Web allows to save 70% of power with respect to the legacy TCP-based architecture. Furthermore, this is not obtained at the cost of a significant degradation in the Quality of Service (QoS). The additional delay introduced by the PS-Web in transferring a Web page is always lower than 1.5 s.

The paper is organized as follows. Section 2 sketches the characteristics of Web traffic. Section 3 is devoted to the definition of the PS-Web architecture. Section 4 reports some experimental results obtained by using the prototype implementation. Finally, Section 5 concludes the paper.

2 Web Traffic Characterization

The power saving strategies implemented in our system are based on the characteristics of the application. Hence, as a preliminary step, it is necessary to understand the traffic profile generated by Web browsing. Many papers in literature provides mathematical characterizations of Web traffic [1,3,4,5,6].

Fig. 1 shows the typical ON/OFF profile of the network traffic generated by an individual Web user [5]. As is well known a Web page consists of a main file and zero or more embedded files (e.g., figures). All files composing a Web page are transferred during the *Active Time* interval while in the *Inactive Time* (or *Think Time*) interval the user reads the content of the downloaded Web page. Within an Active Time, *ON Times* correspond to actual file transfers while during *Active OFF Times* the browser parses a piece of the main file and sends the request for the next embedded file.

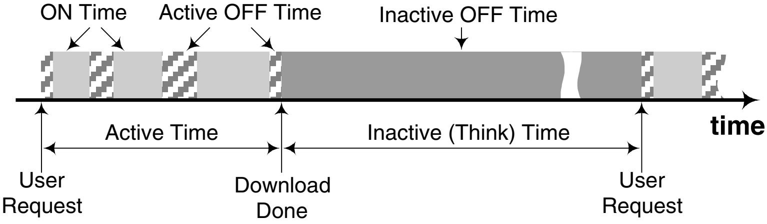


Fig. 1. Typical phases of a page transfer as observed by the user.

Fig. 1 suggests us the following hints. During *Inactive OFF Times* the network interface can be switched off. On the other hand, *Active OFF Times* are often too short (less than 1s) to turn the interface off with some profit (recall that the interface has a transient in going on, during which it consumes energy but is not available for data transfer). However, one could manage the transfer of Web page in such a way that all files in a page are transferred on the wireless link in a single burst. By following this approach different Active OFF Times are concentrated in an single large *OFF Time*, and this gives more chance to turn the interface off, actually saving some energy.

It may be worthwhile to point out that the ON-OFF behavior of Web traffic generated by individual users is related to the self-similarity that is a *structural* property of (aggregate) Web traffic [5]. This means that the ON-OFF behavior is independent of the specific access pattern followed by the user and the type of files available in the Web server.

3 PS-Web Architecture and Protocols

A typical mobile scenario is depicted in Fig. 2. The communication between a mobile host and a machine connected to the Internet (*Fixed Host*) is made possible by a third entity (*Access Point*), which provides Internet connectivity to the mobile host through a wireless link.

Although very simple and costless, a legacy TCP-based solution is prone to various drawbacks that heavily impacts the energy consumption at the mobile host.

1. The TCP congestion control wrongly interprets losses in the wireless link as congestion signals. Hence, the overall throughput is usually low and the wireless network interface at the mobile host remains idle for most of the time.
2. Congestions in the wired networks limits the throughput in the wireless link as well. The overall effect is the same as in 1.
3. The ON/OFF behavior of Web traffic forces the wireless network interface to be inactive for long time intervals.

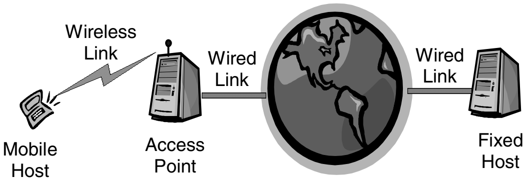


Fig. 2. A typical mobile environment

To overcome these problems we exploited a network architecture based on the *Indirect-TCP* model [2]. The transport connection between the client at the mobile host and the Web server is split into two parts: the first one between the mobile host and the Access Point and the second one between the Access Point and the Web server. At the Access Point an agent (*I-TCP daemon*) relays data from one connection to another. A *Simplified Transport Protocol* (STP in Fig. 3), instead of the legacy TCP protocol, is used to transfer data on the wireless link.

The *Indirect-TCP* model eliminates problems related to point 1 above. However, bottlenecks in the Internet might still cause a low transfer rate in the wireless link. To overcome this second problem, we use pre-fetching of Web pages at the Access Point. Embedded files – if any – are requested to the remote server even without an explicit request from the user and will be transferred to the mobile host, on request from the mobile host itself. This approach allows to transfer embedded files on the wireless link at full speed, irrespective of the throughput available in the wired connection. At the mobile host side, pre-fetching is managed by the PSP (*Power Saving Protocol*) module. At the Access Point side, it is handled by the *PS-Daemon* (see Fig. 3). This is the *I-TCP Daemon* enriched with pre-fetching and power management mechanisms.

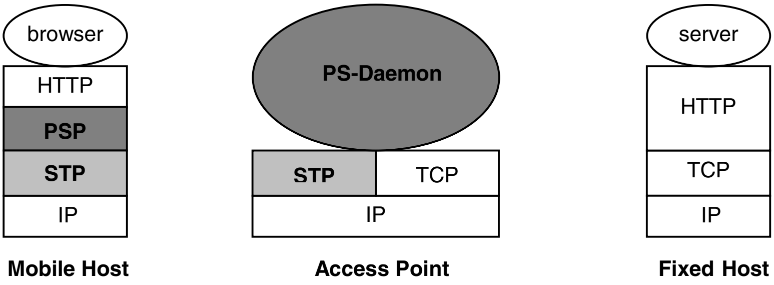


Fig. 3. Overall PS-Web network architecture; evidence on added components

Finally, with reference to point 3 above, it can be observed that, by grouping the transfer of the embedded files on the wireless link in a single burst, Active OFF Times can be compacted in an unique long OFF Time. This reduces significantly the time during which the network interface must be on. At the mobile host, the PSP layer is responsible for identifying the beginning of the Inactive OFF Times, and turning the network interface off until a new request from the browser arrives.

3.1 Power Saving Protocols

The PS-Daemon can be seen as made up of two components. The upper level component interacts with the HTTP modules at the mobile host and fixed server, and implements the same functionalities of a Proxy Web. The lower level component implements power management by interacting with the PSP module at the mobile host via the *Power Saving Protocol.* Therefore, the PS-Daemon can be regarded as a Proxy-Web with power saving support.

Since we are interested in power management, in the following we shall focus on the PSP protocol. Fig. 4 and Fig. 5 show the actions performed at the mobile host and the Access Point, respectively.

Upon receiving the main file from the remote server, the PS-Daemon forwards it to the mobile host, together with an estimate of the residual transfer time (see below), i.e., the time needed to fetch the embedded files from the server (lines 2-3 and 18-22). Upon receiving such an estimate the mobile host turns the network interface off for the corresponding time interval (lines 4-7). Possible requests for embedded files generate by the browser in the meanwhile will be blocked by the PSP layer until the network interface is turned on again (lines 8-12).

When the time interval has elapsed, the PSP module at the mobile host turns the network interface on and sends requests for embedded files, if any, to the PS-Daemon (lines 13-15). The PS-Daemon has already fetched these files from the server and can thus send them back to the browser (lines 23-25).

When the Web page is completely available at the mobile host the PSP module turns the network interface off (lines 16-17) until a new request arrives from the user.

```
 1  OnNewPageRequested(httpRequest)
 2    resumeInterface()
 3    send httpRequest to Access Point
 4    receive (mainFile, estimate) from access point
 5    if(estimate ≥ MIN_USEFUL_TIME)
 6      suspendInterface()
 7    setTimer(estimate)

 8  OnRequestFromBrowser(httpRequest)
 9    if(interface is ON)
10      send httpRequest to Access Point
11    else
12      insert httpRequest into pendingRequests

13  OnTimerExpired()
14    foreach httpRequest in pendingRequests
15      send httpRequest to Access Point

16  OnPageTransferFinished()
17    suspendInterface()
```

Fig. 4. PSP protocol: actions performed at the mobile host.

```
18  OnNewPageRequested(httpRequest)
19    Send httpRequest to server
20    receive mainFile form server
21    estimate = evaluate_time(mainFile)
22    Send (mainFile, estimate) to mobile host

23  OnRequestForEmbedded(httpRequest)
24    File = identifyFile(httpRequest)
25    Send file to mobile host
```

Fig. 5. PSP protocol: actions performed at the Access Point.

The architecture depicted in Fig. 3 is completely transparent to the application and the HTTP protocol, respectively. Like any other Web proxy, the PS-Daemon do not introduce any modification either at the client or server side of the application. In particular, the PSP module at the mobile host presents a socket-like interface to the application layer.

The PS-Web architecture relies upon estimates of the file transfer times. These estimates are performed by the PS-Daemon at the Access Point and communicated to the mobile host (see [17] for details). As it clearly appears from the protocol description, the accuracy of the estimates is a key factor to achieve a significant power saving at the mobile host. The above architecture can be easily modified to include optimizations like handling of inaccuracies in the estimates supplied by the PS-Daemon (line 22, 4), isolation of the application-dependent functionalities to achieve an higher modularity and reusability, and so on. Details on such optimizations can be found in [17].

4 Experimental Results

The objective of our system is twofold. First, it should achieve a good power saving with respect to the legacy approach. At the same time, the reduction in the power consumption should not occur at the cost of an unacceptable degradation in the QoS. To evaluate the performance of our system we considered a *Power Saving Index* (*I_ps*) and a *Page Delay Index* (*I_pd*). The Power Saving Index is defined as

$$I\_ps = \frac{\textit{network interface consumption in PS - Web}}{\textit{network interface consumption in TCP architecture}} \tag{1}$$

I_ps gives an immediate indication of how much energy can be saved by using the PS-Web approach instead of the legacy solution. The Page Delay Index measures the additional delay introduced by the PS-Web to transfer a Web page with respect to the legacy architecture, i.e.,

$$I\_pd = \textit{(page transfer time in PS-Web)} - \textit{(page transfer time in TCP architecture)} \tag{2}$$

To assess our architecture we performed an extensive measurements campaign. In our measurements, to simulate the application level, we used SURGE, a web traffic generator designed by Barford and Crovella [3]. SURGE can simulate an individual user by generating ON/OFF traffic with the same statistical properties of real Web traffic (see Section 2). So, it allows to evaluate our system in realistic conditions.

To significantly simulate a Web session, each experiment stopped after 150 files have been transferred from the web server to the client[1], when the whole page "in flight" arrives at the client. In each experiment, two different instances of a Web client request the same set of pages from the same server. One client uses the PS-Web, while the other one uses the legacy architecture. We take care that the path conditions between the client and the server are the same in both cases by executing the two instances in parallel.

We performed a set of experiments where each set spanned an entire day and was carried out during italian business time to test our architecture when the Internet is heavily loaded. To increase estimates' reliability, each set of experiments was replicated on several days.

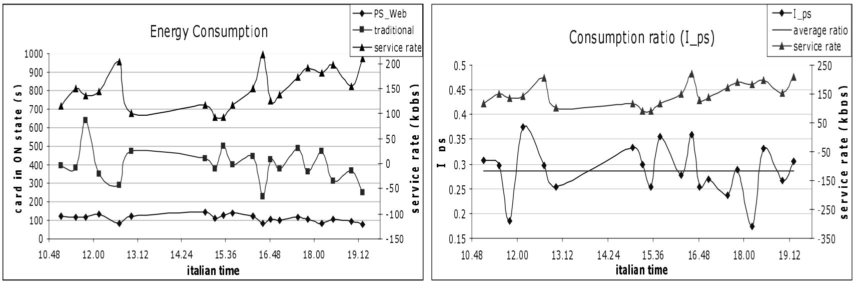
To take into account the influence of the Internet client-server route, we chose to perform separate sets of trials along different paths. Specifically, the mobile host was always located in Pisa (Italy), at the CNUCE-CNR Institute, while the server was located either at EPFL in Lausanne (Switzerland) or at the University of Texas at Arlington.

[1] With SURGE, files requested to the web server are 93% extracted from body and 7% from the tail. Hence, in our experiments, we have about 10 files taken from the tail of the distribution.

4.1 Power Saving Analysis

As shown in [13] and [21], while the network interface is in the on state, it drains a nearly constant power from the battery source, whether it receives or transmits data, or remains idle. The energy consumed by the wireless interface is thus proportional to the time it remains in the on state. Therefore, according to equation (2), *I_ps* can be closely approximated by measuring, both in the legacy and PS-Web architectures, the overall time the network interface is in the on state. This index is easier to compute.

Fig. 6(a) shows the overall time the network interface remains in the on state in the PS-Web and in the legacy architecture, respectively. The available service rate (i.e., the downloading speed in Kbps), as observed by the Web client in the legacy architecture, is also reported. It appears that the PS-Web is almost independent from the service rate. This is due to the joint effect of using the indirect-TCP model and data pre-fetching mechanism: the mobile host turns its network interface on only when the Web page is available at the Access Point. Therefore, the energy consumption does not depend on the state of the connection between the Access Point and the Web server, i.e., the service rate. The small variability that can be observed in Fig. 6(a) is caused by the transfer time estimator (see Section 3).



(a) (b)

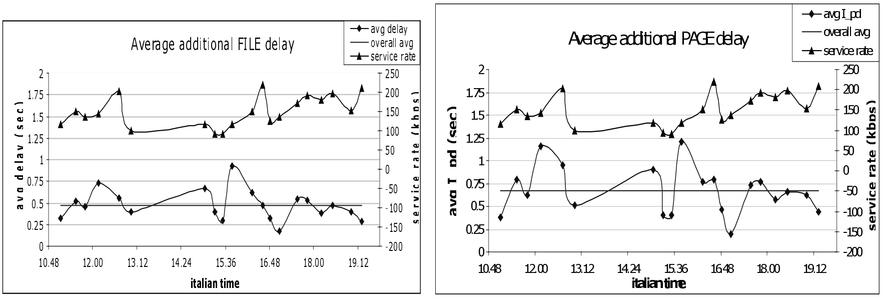
Fig. 6. Energy consumption and *I_ps* as a funcion of time (Web server in Texas).

The above results show that the wireless link is protected from the Internet congestions and, thus, the Indirect-TCP model and data pre-fetching work correctly. This conclusion is corroborated by Fig. 6(b) where the Power Saving Index *I_ps* is reported as a function of the day time. Fig. 6(b) also shows the *I_ps* index averaged on the whole day (average ratio in the figure) and the service rate. The power consumption in the PS-Web is always less than 40% the one in the legacy system, its average value on the whole day is below 30%, with values down to 20%. This means that, by using the PS-Web, we can save always more than 60% of battery, on average more than 70%, with saving peaks over 80%.

The above results refer to the case when the Web server is located in Texas. Analogous experiments have been performed with the Web server in Switzerland, obtaining similar results. Indeed, when the server is in Switzerland the service rate is only slightly higher than the Texas's one (175 Kbps vs. 150 Kbps) and the power saving is slightly lower (68% vs. 71%). The similarity can be justified by observing that the two paths share the initial part, which goes from Pisa to London. Furthermore, in the European business time, when the experiments were performed, this part is more congested than the rest of the path, and thus determine the overall service rate in both cases.

4.2 Delay Analysis

Fig. 7(a) shows the additional delay experienced by Web files, averaged on each experiment and on the whole set of experiments in a day, respectively. The service rate is also included for reader's convenience. Fig. 7(b) shows the same indices but with respect to Web pages instead of Web files (recall that a Web page consists of a main file and possible embedded files).



(a) (b)

Fig. 7. Average additional file and page delay (the Web server is located in Texas).

From Fig. 7(a) it appears that the PS-Web introduces very small additional delays in transferring Web files: the average daily value of the additional delay is less than 0.5 s. Similarly, the additional delay (averaged in a day) related to Web pages is in the order of 0.7 s. In general, the additional page delay is only slightly greater than the additional file delay. This is a very important result since it proves that the prefetching mechanism at the Access Point works properly. In fact, the estimator of pages' transfer times forces the mobile host to keep the network interface off until the Web page is completely available at the Access Point. This means that the first file of a page experiments a certain additional delay, while successive files experiment a smaller one. Therefore, the additional delay related to the Web page is mainly determined by the first file.

Fig. 8 shows the tail of the distribution of the additional delay for files and pages, respectively. The additional delay is never greater than 1s for individual files and never greater than 1.2 s for pages.

Results discussed above were obtained with the Web server located in Texas. However, as above, experiments with the Web server in Switzerland provided very similar results which are omitted for the sake of space. Based on these results we can conclude that the power saving achieved in the PS-Web architecture is not paid with an unacceptable degradation of the QoS preceived by the user.

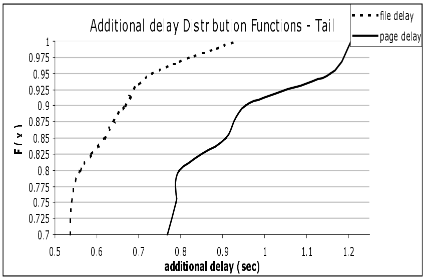


Fig. 8 . Tails of the additional delay distributions, for single files and whole pages

5 Conclusions

In this work we have proposed and experimented new strategies for reducing the power consumption while accessing a Web service in a mobile wireless environment.

To overcome drawbacks caused by the TCP-based legacy architecture we have designed a new architecture – referred to as PS-Web – that implements novel strategies to reduce the power consumption while accessing Web services. Specifically, we operate at the transport layer and we fully exploit information about the application behavior. So, we have an application-dependent approach but our system have been designed in a modular way that can be easily adapted to other network applications, as well. Furthermore, our system requires no modification to the Web application.

The PS-Web architecture is based on the Indirect-TCP model and, as such, it isolates the wireless network from the wired network protecting the former from possible congestions in the latter. Furthermore, it makes a large use of pre-fetching to improve the performance: Web pages are first stored at the Access Point and then downloaded to the mobile host at the maximum rate allowed by the wireless link. Finally, it allows the mobile host to maintain the wireless network interface in the off state during inactivity periods (e.g., user think times).

The experimental results obtained on a prototype implementation of the system have shown that the PS-Web is able to save, on average, the 70% of energy with respect to the legacy TCP-based approach, with saving peaks over 80%. More important, this energy saving is not obtained at the cost of a degradation in the QoS perceived by the user. In fact, experimental results have shown that the additional URT delay in transferring a Web document introduced by the PS-Web with respect to the legacy TCP based approach is, on average, below 1.5 s.

We are currently working on PS-Web in order to refine the estimation of the file transfer time. This is very important since a better estimation allow to minimize the number of times the mobile host's network interface is unnecessarily turned on. Also, we are comparing the performance of our application-dependent approach with those of application-independent solutions that operate without any preliminary information about the application behavior.

Acknowledgements. The authors wish to express their gratitude to Paul Barford for providing the SURGE traffic generator used in the experiments. Also many thanks to Mohan Kumar and Silvia Giordano for giving the opportunity to use Web servers at the University of Texas at Arlington and EPFL (Lausanne, Switzerland), respectively.

References

1. M.Arlitt, C.Williamson, “Internet Web Servers: Workload Characterization and Performance Implication”, *IEEE/ACM Transactions on Networking*, Vol.5, No.5, pp.631-645, Ottobre 1997.
2. A.Bakre, B.R.Badrinath, “Implementation and Performance Evaluation of Indirect TCP”, *IEEE Transactions on Computers*, Vol.46, No.3, Marzo 1997.
3. P.Barford e M.Crovella, “Generating Representative Web Workloads for Network and Server Performance Evaluation”, *Proceedings of ACM SIGMETRICS ′98*, Madison, WI, pp. 151-160, June 1998.
4. P.Barford, A.Bestavros, A.Bradley e M.Crovella, “Changes in Web Client Access Patterns”, to appear in *World Wide Web Journal*, Special Issue on Characterization and Performance Evaluation, 1999.
5. M.Crovella e A.Bestavros, “Self-Similarity in World Wide Web Traffic: Evidence and Possible Causes”, *IEEE/ACM Transaction on Networking*, Vol.5, No.6, pp.835-846, December 1997.
6. C.Cunha, A.Bestavros e M.Crovella, “Characteristics of WWW Client-Based Traces”, Technical Report TR-95-010, Boston Univeristy Department of Computer Science, April 1995.
7. G. H. Forman, J. Zahorjan, "The Challenges of Mobile Computing", *Tecnical Report*, University of Wachington, March 1994.
8. D.P. Helmbold, D.E. Long, B. Sherrod "A Dynamic Disk Spin-down Technique for Mobile Computing", Proceedings of the Second Annual *ACM International Conference on Mobile Computing and Networking*, NY, pp. 130 - 142, November 1996
9. IEEE standard for Wireless LAN- Medium Access Control and Physical Layer Specification, P802.11, November 1997.
10. T. Imielinski, S. Vishwanathan, B.R. Badrinath "Power Efficient Filtering of Data on air", *Proc. of the EDBT*, Cambridge, England, March 1994.
11. T. Imielinscki B.R. Badrinath “Wireless Computing”, *Communication of the ACM*, Vol. 37, No. 10, October 1994.
12. R.Kravets e P.Krishnan, “Power Management Techniques for Mobile Communication”, *Proceedings of the Fourth Annual ACME/IEEE International Conference on Mobile Computing and Networking* (Mobicom’98).
13. G. Anastasi, M. Conti, W. Lapenna, “Power Saving Policies for Wireless Access to TCP/IP Networks”, Proceedings of the *8-th IFIP Workshop on Performance Modelling and Evaluation of ATM and IP Networks (IFIP ATM&IP2000),* Ilkley (UK), July 17-19, 2000.
14. J.R. Lorch, A.J. Smith, “Scheduling Techniques for Reducing Processor Energy Use in MacOS”, *ACM/Baltzer Wireless Networks*, 1997, pp.311-324.
15. J.R.Lorch e A.J.Smith, “Software Strategies for Portable Computer Energy Management”, *IEEE Personal Communication – June 1998*, pp.60-73.
16. M. Othman, S, Hailes, “Power Conservation Strategy for Mobile Computers Using Load Balancing”, *ACM Mobile Computing and Communication Review*, Vol. 2, N. 1, January 1998, pp. 44-50.
17. A. Passarella, “Un’architettura power saving per l’accesso al Web da computer mobile”, *Laurea Thesis*, University of Pisa, October 2001 (in italian).

18. A. Rudenko, P. Reiher, G.J. Popek, G.H. Kuenning, “Saving Portable Computer Battery Power through Remote Process Execution”, *ACM Mobile Computing and Communication Review*, Vol. 2, N. 1, January 1998, pp. 19-26.
19. M.Rulnick e N.Bambos, “Mobile Power Management for Wireless Communication Networks”, *ACM/Baltzer Wireless Networks*, Vol.3, No.1, Marzo 1996.
20. S. Sheng, A. Chandrakasan, R.W. Brodersen, “A Portable Multimedia Terminal”, *IEEE Communications Magazine*, December 1992.
21. M.Stemm e R.H.Katz, “Measuring and Reducing Energy Consumption of Network Interfaces in Hand-Held Devices”, *Proc. 3° International Workshop on Mobile Multimedia Communication*, Princeton, NJ, Settembre 1996.
22. Mark Weiser, Brent Welch, Alan Demers Scott Shenker. "Scheduling for Reducing CPU Energy", USENIX Association, *First Symposium on Operating System Design and Implementation*, Monterey, CA, Nov. 1994.
23. M.Zorzi e R.R.Rao, “ARQ Error Control on Fading Mobile Radio Channels”, accepted for pubblication in *IEEE Trans. Veh. Tech.*, Also in *Proc. IEEE ICUPC ’95*, pp.211-215, Novembre
24. M.Zorzi e R.R.Rao, “Energy Constrained Error Control for Wireless Channels”, *Proceeding of IEEE GLOBECOM ’96*, pp.1411-1416, 1996.

A Resource/Connection Management Scheme for HTTP Proxy Servers

Takuya Okamoto[1], Tatsuhiko Terai[1], Go Hasegawa[2], and Masayuki Murata[2]

[1] Graduate School of Engineering Science, Osaka University
1-3 Machikaneyama, Toyonaka, Osaka 560-8531, Japan
{tak-okmt, terai}@ics.es.osaka-u.ac.jp

[2] Cybermedia Center, Osaka University
1-30 Machikaneyama, Toyonaka, Osaka 560-0043, Japan
{hasegawa, murata}@cmc.osaka-u.ac.jp

Abstract. Although many research efforts have been devoted to the network congestion against an increase of the Internet traffic, there has been a little concern on improvement of the performance of Internet hosts in spite of the projection that the bottleneck is now being shifted from the network to hosts. We have proposed SSBT (Scalable Socket Buffer Tuning), which is intended to improve the performance of Web servers by maintaining their resources effectively and fairly, and validated its effectiveness through the simulation and implementation experiments. In the current Internet, however, a significant amount of Web document transfer requests are through HTTP proxy servers. Accordingly, in this paper, we propose a new resource management scheme for proxy servers to improve their performance and to reduce Web document transfer time via the proxy servers. Our proposed scheme has the following two components. One is an enhanced E-ATBT, which is an enhancement version of our previous SSBT for proxy servers by taking account of different characteristics among TCP connections. The other is a scheme that manages persistent TCP connections at proxy servers to avoid newly arriving TCP connections from being rejected due to lack of resources. We validate an effectiveness of our proposed scheme through simulation experiments, and confirm that it can manage proxy server resources effectively.

1 Introduction

With the rapid growth of Internet users, many research efforts have been directed to avoiding and dissolving network congestion against an increase of network traffic. However, there has been a little concern on improvement of the performance of Internet hosts in spite of the projection that the performance bottleneck is now being shifted from the network to endhosts.

In [1], we have proposed SSBT (Scalable Socket Buffer Tuning) which is intended to improve the performance of Web servers by maintaining their resources effectively and fairly. SSBT has two major components; E-ATBT (Equation-based Automatic TCP Buffer Tuning) and SMR (Simple Memory-copy Reduction) schemes. In E-ATBT, we maintain an 'expected' throughput value of each active TCP connection, which is determined by an analytic estimation [2]. It is characterized by packet loss ratio, RTT (Round

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 252–263, 2002.

Trip Time), and RTO (Retransmission Time Out), which are easily monitored by a sender host. Then, the send socket buffer is assigned to each connection based on its expected throughput with consideration on a max-min fairness among connections. The SMR scheme provides a set of socket system calls in order to reduce the number of memory copy operations at the sender host in TCP data transfer. The SMR scheme is alike as other schemes [3,4], but it is simpler to implement.

In the current Internet, there are many requests for Web documents transfer via HTTP proxy servers [5]. Since the proxy servers are usually prepared by ISPs (Internet Service Providers) for their customers, such proxy servers must accommodate a large number of the customers' HTTP accesses simultaneously. Furthermore, the proxy servers should handle both of upward TCP connections (from the proxy server to Web servers) and downward TCP connections (from the client hosts to the proxy server). Therefore, it is likely that the proxy server becomes the bottleneck in Web document transfer, even when both of the network bandwidth and the Web server performance are large enough. That is, to reduce the Web document transfer time, a performance enhancement of the proxy servers should next be considered.

In this paper, we first point out several problems in handling TCP connections at the HTTP proxy server. The one is the assignment of the socket buffer for TCP connections at the proxy server. When a TCP connection is not assigned the proper size of send/receive socket buffer according to its throughput, the assigned socket buffer may be left unused or insufficient, which results in waste of the socket buffer. Another problem is the management of persistent TCP connections, which tends to waste the resource of the busy proxy server. When a proxy server accommodates many persistent TCP connections without any effective management, its resources are kept assigned to those connections whether those connections are actually 'active' or not. Then new TCP connections cannot be established since the server resources are short.

We propose a new resource management scheme for proxy servers to resolve such problems, and then to reduce Web document transfer time via the proxy servers. Our proposed scheme has following two features. One is an enhanced E-ATBT, which is an enhancement version of our previous E-ATBT for proxy servers. Differently from the Web servers, the proxy server should handle both upward and downward TCP connections and behave as a client host to obtain Web documents and in-line images from Web servers. We therefore enhance E-ATBT to effectively handle a dependency between upward and downward TCP connections and to assign its receive socket buffer size dynamically. The other is a resource management scheme that can avoid newly arriving TCP connections from being rejected due to lack of resources for establishing them on the proxy server. It involves the management of persistent TCP connections provided by the HTTP/1.1. The persistent connection can omit the overhead of TCP's three-way handshake and then reduce the document transfer time by HTTP. However, when the persistent TCP connection is unused until it is closed by timeout mechanism, the resources assigned for the TCP connection are wasted. The proposed scheme intentionally tries to close the persistent connections when the resources of the proxy server are shorthanded.

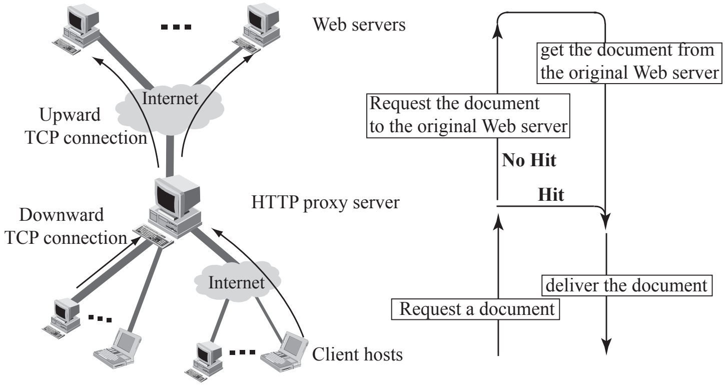


Fig. 1. HTTP Proxy Server

2 Background

2.1 Proxy Server

An HTTP proxy server works as an agent for Web client hosts that request Web documents. When it receives Web document transfer requests from the Web client host, it obtains the requested document from the original Web servers on behalf of the client host and delivers it to the client. It also caches obtained Web documents. When other client hosts request the same documents, it transfers the cached documents, which results in that the document transfer time is much reduced. For example, it is reported in [6] that using Web proxy servers reduces document transfer time by up to 30%. Also, when the cache is hit, the document transfer is performed without any connection establishment to Web servers. Thus, the congestion within the network and at Web servers can also be reduced.

The proxy server accommodates a large number of connections, which are connected from Web client hosts and to Web servers as depicted in Figure 1. It is a different point from Web servers. The proxy server behaves as a sender host for the downward TCP connection (between the client host and the proxy server) and as a receiver host for the upward TCP connection (between the proxy server and the Web server). Therefore, if the resource management is not appropriately configured at the proxy server, the document transferring time increases even when the network is not congested or the load at the Web server is not high. That is, careful and effective resource management is a critical issue for improving the performance of the proxy server. In the current Internet, however, most proxy servers including those in [7,8] are lack of such considerations.

Resources of HTTP proxy servers that we focus in this paper are *mbuf*, *file descriptor*, *control blocks*, and *socket buffer*. Those are closely related to the performance of TCP connections in transferring Web documents. Mbuf, file descriptor, and control blocks are resources for TCP connections. The amount of those resources cannot be changed dynamically according to the requirement of the proxy server, since it is determined when the system kernel is booted or when the proxy server is activated [9]. When at least one of the resources lacks, therefore, newly arriving TCP connections for Web document transfer have to wait for other connections to be closed and their assigned resources to be released.

The socket buffer is used for data transfer operations between user applications and the sender/receiver TCP. When the user application transmits data using TCP, the data is copied to the send socket buffer and subsequently it is copied to the mbufs (or mbuf clusters). The size of the assigned socket buffer is a key issue for the effective data transfer by TCP. Suppose that a server host is sending TCP data to two client hosts; one a 64 Kbps dial-up (say, client A) and the other a 100 Mbps LAN (client B). If the server host assigns equal size of send socket buffers to both client hosts, it is likely that the amount of the assigned buffer is too large for client A and too small for client B, because of the differences of capacity (more strictly, bandwidth-delay products) of their connections. For an effective buffer allocation to both client hosts, a compromise of the buffer usage should be taken into account.

We proposed an E-ATBT scheme [1], which assigns the receive socket buffer to each TCP connection dynamically according to its throughput estimated from the observed network parameters, such as packet loss ratio, RTT, and RTO. That is, a sender host calculates the average window size of its TCP connection based on the analysis result in [10] from the above three parameters. The throughput of the TCP connection is then obtained by considering the performance degradation caused by TCP's retransmission timeout. Finally, we estimate the required receive socket buffer size as multiplication of the estimated throughput and RTT of the TCP connection. By taking into account the observed network parameters, the resource at the Web server is appropriately allocated to connections in various network environments.

E-ATBT is applicable to HTTP proxy servers, since the proxy servers also accommodate many TCP connections issued by clients in various environments. However, since proxy servers have a *dependency* between upward and downward TCP connections, a straightforward application of E-ATBT is insufficient. Furthermore, the proxy server behaves as a receiver host for the upward TCP connection to the Web server, we have to consider the management scheme for the receive socket buffer, which was not considered in the original E-ATBT.

2.2 Persistent TCP Connection of HTTP/1.1

In recent years, many Web servers and client hosts (namely, Web browsers) support a *persistent connection* option, which is one of the important functions of HTTP/1.1 [11]. In the older version of HTTP (HTTP/1.0), the TCP connection between server and client hosts is immediately closed when the document transfer is completed. However, since Web documents have many in-line images, it is necessary to establish TCP connections many times to download them in HTTP/1.0. It results in a significant increase of document transfer time since the average size of Web documents at several Web servers is about 10 [KBytes] [12,13]. The three-way handshake in each TCP connection establishment makes the situation worse.

In the persistent connection of HTTP/1.1, on the other hand, the server preserves the status of the TCP connection, which includes the congestion window size, RTT, RTO, ssthresh, and so on, when it finishes the document transfer, and re-uses the connection and its status when other documents are transferred by using the same HTTP session. Then, the three-way handshake can be omitted. However, since it keeps the TCP connection established whether the connection is active (in use for packet transfer) or not, the resources at the server are wasted when the TCP connection is inactive. Therefore, the significant portion of the resources may be wasted in order to keep the persistent TCP connections at the proxy server accommodating many TCP connections.

One solution against this problem is simply to discard HTTP/1.1 and to use HTTP/1.0, since HTTP/1.0 closes the TCP connection when the document transfer is finished. However, HTTP/1.1 has other elegant mechanisms such as the pipelining and the contents negotiation [11]. We should therefore develop an effective resource management scheme under HTTP/1.1. Our solution is that the proxy server aggressively closes the persistent TCP connections that are unnecessarily wasting the proxy resources, as the resources become short.

3 Algorithm and Implementation Issues

In this section, we propose a new resource management scheme suitable to the HTTP proxy server, which consists of a new management scheme of send/receive socket buffer, and a handling algorithm of persistent TCP connections.

3.1 New Socket Buffer Management Method

Handling the Relation of Upward and Downward Connections

A HTTP proxy server relays a document transfer request to a Web server for a Web client host. Thus, there is a close relation between an upward TCP connection (from the proxy server to the Web server) and a downward TCP connection (from the client to the proxy server). That is, the difference of the throughput of both connections should be taken into account when socket buffers are assigned to them. For example, when the throughput of a certain downward TCP connection is larger than that of other concurrent downward TCP connections, the larger size of socket buffer should be assigned to the TCP connection by using E-ATBT. However, if the throughput of the upward TCP connection corresponding to the downward TCP connection is low, the send socket buffer assigned to the downward TCP connection is likely not to be fully utilized. In this case, the unused send socket buffer should be assigned to the other concurrent TCP connections having smaller socket buffers, hence, that the throughputs of those TCP connections would be improved.

There is one problem to realize the above-mentioned method. The TCP connection is identified with the control blocks, `tcpcb`, by the kernel. However, the relation between the upward and downward connections cannot be known by the kernel. Two possible ways to overcome this problem are considered as follows:

- The proxy server monitors the utilization of the send socket buffer of downward TCP connections. Then, it decreases the assigned buffer size of connections whose send socket buffers are not fully utilized.
- When the proxy server sends the document transfer request to the Web server, the proxy server attaches an information of the relation to the packet header.

The former algorithm can be done only by the modification of the proxy server. On the other hand, the latter algorithm needs the interaction of the HTTP protocol. In the higher abstract model, the above two algorithms have a same effect. However, the latter has a implementation difficulty while it can achieve a precise control.

Control of Receive Socket Buffer

In most of past researches, it was assumed that a receiver host has enough large size of receive socket buffer, considering that the performance bottleneck of the data transfer is not at the endhosts, but within the network. Therefore, many OSs assign a small size of the receive socket buffer to each TCP connection. For example, the default size of the receive socket buffer in the FreeBSD system is 16 [KBytes]. Now it is very small [14] because the network bandwidth is dramatically increased in the current Internet, and the performance of the Internet servers becomes higher and higher.

To avoid the performance limit by the receive socket buffer, the receiver host should adjust its receive socket buffer size to the congestion window size of the sender host. This can be done by monitoring the utilization of the receive socket buffer, or by adding information about the window size to data packet header, as described above. In the simulation in the next section, we suppose that the proxy server can obtain complete information about required sizes of the receive socket buffer of upward TCP connections and control them according to the required size.

3.2 Connection Management

As explained in Subsection 2.2, a careful treatment of persistent TCP connections on the proxy server is necessary for an effective usage of the resources of the proxy server. We propose a new management scheme of persistent TCP connections at the proxy server by considering the amount of the remaining resources. The key idea is as follows. When the load at the proxy server is low and the remaining resources are much enough, it tries to keep as many TCP connections as possible. On the other hand, when the resources at the proxy server are going to be short, the proxy server tries to close the persistent TCP connections and free the resources, such that the released resources can be used for new TCP connections.

The remaining resources of proxy servers should be monitored for realizing the above-mentioned control method. The resources for establishing TCP connections at the proxy server include *mbuf*, *file descriptor*, and *control blocks*. The total amount of these resources cannot be changed dynamically after the kernel is booted. However, the total and remaining amounts of these resources can be observed in kernel system [9]. Therefore, we introduce threshold values of the utilization for these resources, and if one of utilization level of those resources, calculated by the kernel system at regular intervals, reaches the threshold, the proxy server starts closing the persistent TCP connections and releasing the resources assigned to the connections.

The proxy server maintains persistent TCP connections as follows. See also Figure 2. When a TCP connection becomes idle, the proxy server records the connection and the current time. For fast lookup of the record, we use the hashing algorithms, of which key is the combination of source/destination IP addresses and the port number of the TCP connection. We also introduce a list, called a *time scheduling list*, to put the persistent connections in order of the time length that they are persistent. When a new persistent TCP connection is registered hash table, it is added at the end of the time scheduling list, so that the proxy server can select the older persistent TCP connections to be closed.

Each entry in the hash table has the socket file descriptor of the corresponding to the TCP connection, which is used later to identify the connection. When the proxy

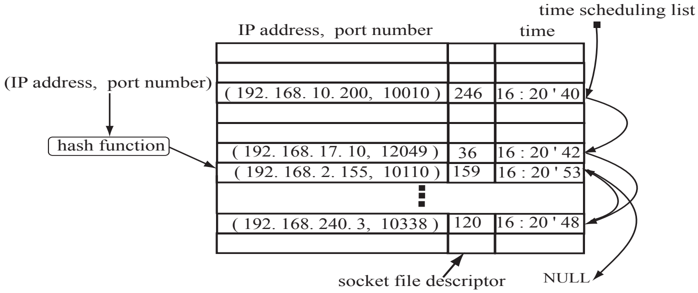


Fig. 2. Management Scheme of Persistent TCP Connections

server closes some of persistent TCP connections, it selects them from the top of the time scheduling list, by which the proxy server can close the older persistent connections. When a certain persistent TCP connection in the hash table becomes active before closed, or when it is closed by persistent timer expiration, the proxy server removes the corresponding entry from the hash table and the time scheduling list. All operations on the persistent TCP connections can be performed by simple pointer manupilations and hash operations.

For the further effective resource usage, we also add the mechanism that the amount of resources assigned to the persistent TCP connections is decreased gradually after the connection is inactive. The socket buffer is not necessary at all when the TCP connection becomes idle. However, we gradually decrease the send socket/receive buffer size of persistent TCP connections by taking account of the fact that as the connection idle time continues, the possibility that the TCP connection is ceased becomes large.

4 Simulation Experiments

In this section, we investigate the effectiveness of our proposed scheme through simulation experiments using ns-2 [15]. Figure 3 shows the simulation model. In this figure, the bandwidths of the links between client hosts and an HTTP proxy server and those between the proxy server and Web servers are all set to 100 Mbps. To see the effect of various network conditions, we set the packet loss probability on each link to be 0.0001, 0.0005, 0.001, 0.005 or 0.01. That is, one-fifth of the links is assigned one of the above values. The propagation delay of each link between the client hosts and the proxy server is also varied as ranged from 10 msec and 100 msec, and that between the proxy server and the Web servers is from 10 msec and 200 msec. The propagation delays of each link is determined randomly from the above ranges. The number of Web servers is fixed at 50, and that of the client hosts is changed as 50, 100, 200 and 500. We ran 300 sec simulation in each experiment.

In the simulation experiments, each client host selects one of the Web servers at random and generates a document transfer request via the proxy server. The distribution of the requested document size is obtained from [12], which is given by the combination of a log-normal distribution for small documents and a Pareto distribution for large

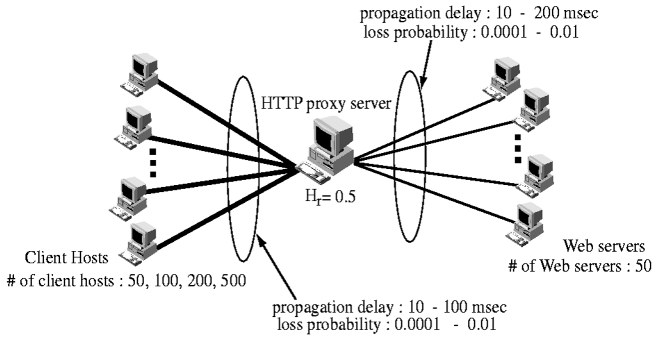

propagation delay : 10 - 200 msec
loss probability : 0.0001 - 0.01
HTTP proxy server
H_r= 0.5
Client Hosts
# of client hosts : 50, 100, 200, 500
Web servers
# of Web servers : 50
propagation delay : 10 - 100 msec
loss probability : 0.0001 - 0.01

Fig. 3. Simulation Model

ones. Note that since we focus on the resource and connection management of proxy servers, we have not considered detailed algorithms of the caching behavior, including the cache replacement algorithms. Instead, we set the hit ratio, H_r, to 0.5. Using H_r, the proxy server decides either to transfer the requested document to the client directly, or to deliver it to the client after downloading it from the Web server. The proxy server has 3200 KBytes of socket buffer and assigns it as send/receive socket buffer to TCP connections. It means that the original scheme can establish at most 200 TCP connections concurrently, since it fixedly assigns 16 KBytes of send/receive socket buffer to each TCP connection.

In what follows, we compare the performance of the following 4 schemes; scheme (1) which does not use any enhanced algorithms in this paper, scheme (2) which uses E^2-ATBT, scheme (3) which uses E^2-ATBT and the connection management scheme described in Subsection 3.2, and scheme (4) which uses E^2-ATBT and the connection management scheme with the algorithm that gradually decreases the socket buffer assigned to the persistent TCP connections. Note that for scheme (3) and (4), we do not explicitly consider the amount and the threshold value of each resource, as explained in Subsection 3.2. Instead, we introduce N_{max}, the maximum number of connections which can be established simultaneously, to simulate the limitation of the proxy server resources. In scheme (1) and (2), newly arrived requests are rejected when the number of TCP connections in the proxy server is N_{max}. On the other hand, scheme (3) and (4) forcibly terminate some of persistent TCP connections that are unused for the document transfer, and establish the new TCP connections. For scheme (4), we exclude persistent TCP connections from calculation process of E^2-ATBT algorithm, and halve the assigned size of socket buffer every 3 sec. The minimum size of the socket buffer is 1 KByte.

4.1 Evaluation of Proxy Server Performance

We first investigate the performance of the proxy server. Here we define the performance of proxy server as the total size of the documents transferred in both directions by the proxy server during 300 sec simulation time. In Figure 4, we plot the performance of

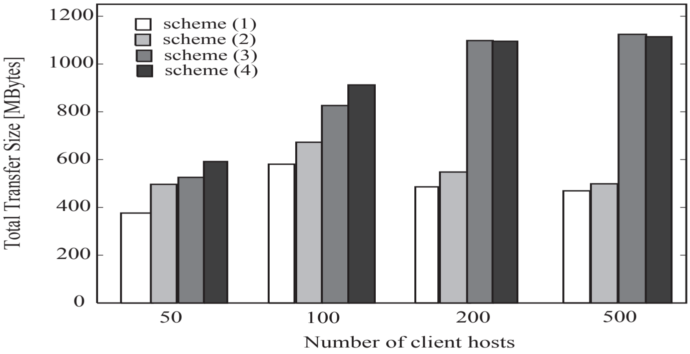
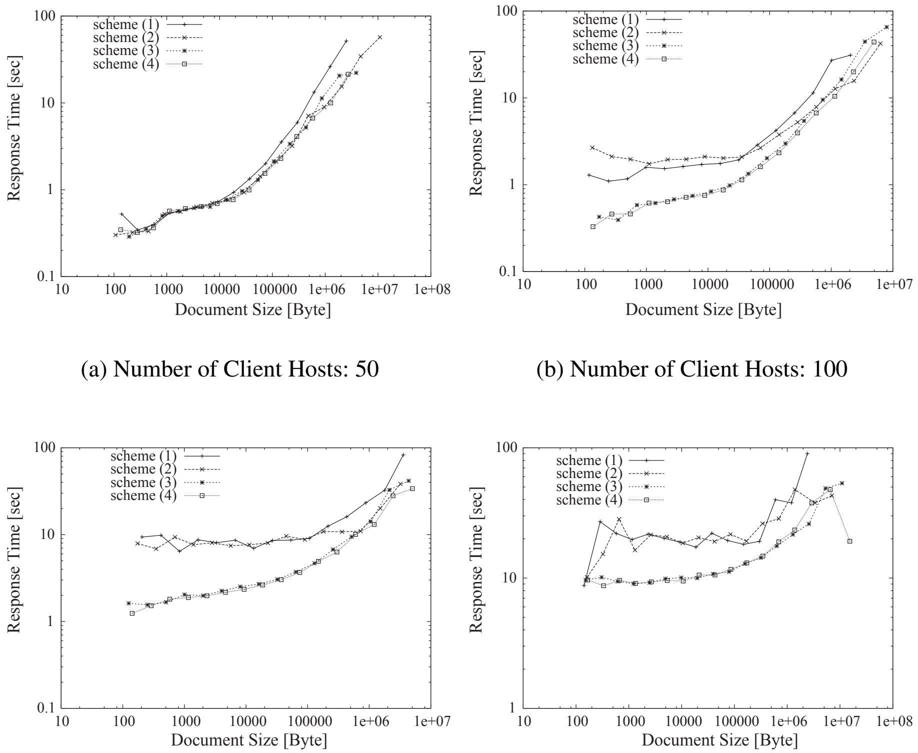


Fig. 4. Simulation Result: Proxy Server Performance

the proxy server as a function of the number of client hosts. Here, we set N_{max} to 200. It is clear from this figure that the performance of the original scheme (scheme (1)) is decreased in the case of the larger number of client hosts. It is because when the number of client hosts is larger than N_{max}, the proxy server rejects some of document transfer requests, although most of N_{max} TCP connections are idle, which means that they do nothing but waste the resources of the proxy server. The results of scheme (2) in Figure 4 shows that E^2-ATBT can improve the proxy server performance regardless of the number of client hosts. However, it also shows that the performance also degrades when the number of client hosts increases. This means that E^2-ATBT cannot solve the problem of ‘idle’ persistent TCP connections, and that it is necessary to introduce a connection management scheme to overcome this problem.

We can also see that scheme (3) can significantly improve the performance of the proxy server, especially when the number of client hosts is large. It is since when the proxy server cannot accept all connections from the client hosts, which corresponds to the case where the number of client hosts is larger than 200 in Figure 4, scheme (3) would close idle TCP connections for newly arriving TCP connections to be established. It results in that the number of TCP connections which actually transfer documents increases largely. Scheme (4) can also improve the performance of the proxy server, especially when the number of client hosts is small, as shown in Figure 4. In the case of larger number of client hosts, however, there is little performance improvement. It can be explained as follows. When the number of client hosts is small, most of the persistent TCP connections at the proxy server are kept established, since the proxy server has enough resources to accommodate 50 client hosts. Therefore, the socket buffer assigned to the persistent TCP connections can be effectively re-assigned to other active TCP connections by scheme (4). When the number of client hosts is large, on the other hand, the persistent TCP connections are likely to be closed before scheme (4) begins to decrease the assigned socket buffer. It results in that scheme (4) can do nothing against the persistent TCP connections.



(a) Number of Client Hosts: 50

(b) Number of Client Hosts: 100

(c) Number of Client Hosts: 200

(d) Number of Client Hosts: 500

Fig. 5. Simulation Result: Response Time

4.2 Evaluation of Response Time

We next show the evaluation results of response time of document transfer, which corresponds to the user-perceived performance. We define the response time as the time from when a client host sends a document transfer request to when it receives the requested document. It also includes the waiting time for connection establishment. Figure 5 shows the simulation results. We plot the response time as a function of document size for the four schemes. From this figure, we can clearly observe that the response time is much improved when our proposed scheme is applied especially when the number of connections is large (Figure 5 (b)-(d)). However, when the number of client hosts is 50, the proposed scheme does not help improving the response time. For this, the server resources are enough to accommodate 50 client hosts and all TCP connections are soon established at the proxy server. Therefore, response time can not be improved so much. Note that since E^2-ATBT can improve the throughput of TCP data transfer to some degree, the proxy server performance can be improved as shown in the previous subsection.

Although schemes (3) and (4) can improve the response time largely, there is little difference between the two schemes. This can be explained as follows. Scheme (4) decreases the assigned socket buffer to persistent TCP connections and re-assign it to other active TCP connections. Although the throughput of the active TCP connections becomes improved, its effect on the response time is very small compared with the effect of introducing scheme (3). However, scheme (4) is worth to be used at the proxy server, since scheme (4) can give a good effect on the proxy server performance as shown in Figure 3.

From all of the above simulation results, we can say that scheme (4), which has all enhanced mechanisms proposed in this paper, is the best one to improve both the performance of the proxy server and response time of client hosts, regardless of the number of client hosts.

5 Conclusion

In this paper, we have proposed a new resource management scheme for HTTP proxy servers. Our proposal scheme has two algorithms. One is an enhanced E-ATBT, the scheme for managing the socket buffer considering about the relation between the upward and downward TCP connections, which is one of the characteristics of the proxy servers. It also manages the receiver socket buffer, which is not considered in the original E-ATBT. The other is the scheme for managing TCP connections at the proxy servers. It maintains persistent TCP connections, and it aggressively closes them when the resources lack. We have evaluated our scheme through some simulation experiments, and confirmed that our scheme can improve the performance of the proxy servers, and reduce document transfer time experienced by client hosts.

We are now implementing the proposed scheme to the actual proxy server, and to evaluate it through experiments using the actual network. We also plan to introduce other kinds of the resources of Web servers and proxy servers to our resource management scheme. For example, a CPU processing time should be considered for executing CGI programs, which is one of the bottleneck of the busy Web servers.

Acknowledgements. This work was partly supported by the Research for the Future Program of the Japan Society for the Promotion of Science under the Project "Integrated Network Architecture for Advanced Multimedia Application Systems," Telecommunication Advancement Organization of Japan under the Project "Global Experimental Networks for Information Society Project," and the "Research on High-performance WWW server for the Next-Generation Internet" program of from the Telecommunications Advancement Foundation.

References

1. G. Hasegawa, T. Terai, T. Okamoto, and M. Murata, "Scalable socket buffer tuning for high-performance Web servers," in *Proceedings of IEEE ICNP 2001*, Nov. 2001.
2. G. Hasegawa, T. Matsuo, M. Murata, and H. Miyahara, "Comparisons of packet scheduling algorithms for fair service among connections on the internet," in *Proceedings of IEEE INFOCOM 2000*, Mar. 2000.

3. A. Gallatin, J. Chase, and K. Yocum, "Trapeze/IP: TCP/IP at near-gigabit speeds," in *Proceedings of 1999 USENIX Technical Conference*, June 1999.
4. P. Druschel and L. Peterson, "Fbufs: A high-bandwidth cross-domain transfer facility," in *Proceedings of the Fourteenth ACM symposium on Operating Systems Principles*, pp. 189–202, Dec. 1993.
5. Proxy Survey, available at `http://www.delegate.org/survey/proxy.cgi`.
6. A. Feldmann, R. Caceres, F. Douglis, G. Glass, and M. Rabinovich, "Performance of Web proxy caching in heterogeneous bandwidth environments," in *Proceedings of IEEE INFOCOM '99*, pp. 107–116, 1999.
7. Squid Home Page, available at `http://www.squid-cache.org/`.
8. Apache proxy `mod_proxy`, available at
   `http://httpd.apache.org/docs/mod/mod_proxy.html`.
9. M. K. McKusick, K. Bostic, M. J. Karels, and J. S. Quarterman, *The Design and Implementation of the 4.4 BSD Operating System.* Reading, Massachusetts: Addison-Wesley, 1999.
10. J. Padhye, V. Firoiu, D. Towsley, and J. Krusoe, "Modeling TCP throughput: A simple model and its empirical validation," in *Proceedings of ACM SIGCOMM '98*, pp. 303–314, Aug. 1998.
11. R. Fielding, J. Gettys, J. Mogul, H. Frystyk, and T. Berners-Lee, "Hypertext transfer protocol – HTTP/1.1," *Request for Comments (RFC) 2068*, Jan. 1997.
12. P. Barford and M. Crovella, "Generating representative Web workloads for network and server performance evaluation," in *Proceedings of the 1998 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, pp. 151–160, July 1998.
13. M. Nabe, M. Murata, and H. Miyahara, "Analysis and modeling of World Wide Web traffic for capacity dimensioning of Internet access lines," *Performance Evaluation*, vol. 34, pp. 249–271, Dec. 1999.
14. M. Allman, "A Web server's view of the transport layer," *ACM Computer Communication Review*, vol. 30, pp. 10–20, Oct. 2000.
15. The VINT Project, "UCB/LBNL/VINT network simulator - ns (version 2)." available at `http://www.isi.edu/nsnam/ns/`.

Measurement-Based Modeling of Internet Round-Trip Time Dynamics Using System Identification

Hiroyuki Ohsaki[1], Mitsushige Morita[2], and Masayuki Murata[1]

[1] Cybermedia Center, Osaka University
1-30 Machikaneyama, Toyonaka, Osaka, Japan
{oosaki, murata}@cmc.osaka-u.ac.jp

[2] Graduate School of Engineering Science, Osaka University
1-3 Machikaneyama, Toyonaka, Osaka, Japan
m-morita@ics.es.osaka-u.ac.jp

Abstract. Understanding the end-to-end packet delay dynamics of the Internet is of crucial importance since it directly affects the QoS (Quality of Services) of various applications, and it enables us to design an efficient congestion control mechanism. In our previous studies, we have measured round-trip time of the Internet, and have modeled its dynamics by the ARX (Auto-Regressive eXogenous) model using system identification. As input and output data for the ARX model, we have used the packet inter-departure time from a source host and the corresponding round-trip time variation measured by the source host. In the current paper, for improving the model accuracy, we instead use the packet transmission rate from the source host and the average round-trip time measured by the source host. Using input and output data measured in working LAN and WAN environments, we model the round-trip time dynamics by determining coefficients of the ARX model using system identification. Through numerical examples, we show that in LAN environment, the round-trip time dynamics can be accurately modeled by the ARX model. We also show that in WAN environment, the round-trip time dynamics can be accurately modeled when the bottleneck link is shared by a small number of users.

1 Introduction

In the past decade, the Internet has been explosively growing in scale as well as in population after the introduction of the WWW (World Wide Web). In January 1997, only 16 million computers were connected to the Internet, but it has jumped to more than 56 million computers in July 1999 [1]. Because of the changing nature of the Internet, nobody knows the current network topology of the Internet. Such uncertainty of the Internet makes it very difficult, but also challenging, to analyze and understand the end-to-end packet behavior of the Internet.

Understanding the end-to-end packet delay dynamics of the Internet is of crucial importance since (1) it directly affects the QoS (Quality of Services) of various applications, and (2) it enables us to design an efficient congestion control mechanism for both realtime and non-realtime applications. For non-realtime applications, a delay-based approach for congestion control mechanisms, rather than a loss-based approach as used in

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 264–276, 2002.

TCP (Transmission Control Protocol), has been proposed (e.g., [2,3]). The main advantage of such a delay-based approach is, if it is properly designed, packet losses can be prevented by anticipating impending congestion from increasing packet delays.

In [4,5], we have proposed a novel approach for modeling the end-to-end packet delay dynamics of the Internet using system identification. In [4,5], we have regarded the network, seen by a specific source host, as a dynamic SISO (Single-Input and Single Output) system. We have modeled the round-trip time dynamics using the ARX (Auto-Regressive eXogenous) model. In those studies, the input to the system was the packet inter-departure time from the source host, and the output was the round-trip time variation between two adjacent packets. Using measured data obtained in wired and wireless LAN environments, we have investigated how accurately the ARX model can capture the round-trip time dynamics of the Internet. We have found that the ARX model can capture the round-trip time dynamics when the network is moderately congested. We have also found that, when the network is not congested or the measured round-trip time is noisy, the ARX model fails to capture the dynamics.

This paper is a direct extension of [4,5], and has three major changes: (1) refined definition of input and output data for improving the model accuracy, (2) experimentations in LAN and WAN environments, and (3) use of two model validation methods in time domain and frequency domain. The first change is to refine the definition of the input and the output for the ARX model. The input to the system is changed to an *instantaneous* packet transmission rate from the source host during a fixed sampling interval. Also the output is changed to an *instantaneous* average round-trip time observed by the source host during a fixed sampling interval. In [4,5], the sampling interval is not fixed since it is dependent on the packet sending/receiving process at the source host. On the contrary, in this paper, the sampling interval is fixed, so that the model accuracy is expected to be improved since system identification originally assumes a fixed sampling interval. The objective of the second change is to investigate how the model accuracy is related to a network configuration. We collect input and output data for system identification in LAN and WAN environments, and build a model for the round-trip time dynamics. The third change is to evaluate the model accuracy in a more rigorous manner. We evaluate the model accuracy in frequency domain as well as in time domain. In [5], the accuracy of the ARX model was evaluated only in time domain; that is, we have compared the simulated outputs from the ARX model (i.e., round-trip times) with the actual round-trip times. In this paper, we also examine the model accuracy in frequency domain using a spectral analysis. Through numerical examples, we show that in LAN environment, the round-trip time dynamics can be accurately modeled by the ARX model. We also show that in WAN environment, the round-trip time dynamics can be accurately modeled when the bottleneck link is shared by a small number of users.

This paper is organized as follows. In Section 2, a black-box approach for modeling the round-trip time dynamics of the Internet is explained. In Section 3, we discuss several measurement methods of the round-trip time, in particular, for collecting input and output data for system identification. We also explain three network environments in which input and output data, used for the model identification and for the model validation, are collected. Section 4 shows several measurement and modeling results, and discuss how accurately the ARX model can capture the round-trip time dynamics in various network configurations. Section 5 concludes this paper with a few remarks.

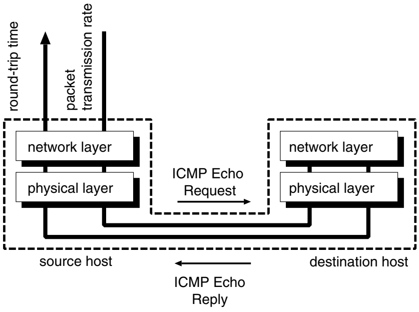


Fig. 1. Modeling round-trip time dynamics as SISO system.

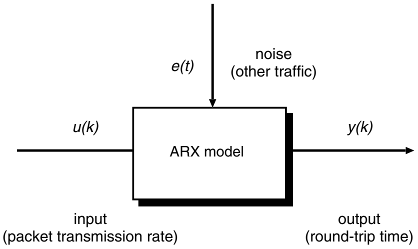


Fig. 2. ARX model for modeling round-trip time dynamics.

2 Black-Box Modeling Using ARX Model

As depicted in Fig. 1, the network seen by a specific source host, including underlying protocol layers (e..g, physical, data-link, and network layers), is considered as a black-box. Our goal of this paper is to model a SISO system describing the round-trip time dynamics: i.e., the relation between a packet sending process from the source host and its resulting round-trip time observed at the source host. Effects of other traffic (i.e., packets coming from other hosts) are modeled as *noise*. As the input to the system, we use an *instantaneous* packet transmission rate from the source host: i.e., the packet transmission rate during a fixed sampling interval. As the output from the system, we use an *instantaneous* average round-trip time measured by the source host: i.e., the average round-trip time during a fixed sampling interval.

In this paper, the ARX model is used and its coefficients are determined using system identification [6]. Figure 2 illustrates a fundamental concept of using the ARX model for capturing the round-trip time dynamics. The input to the ARX model is a packet transmission rate from the source host, and the output from the ARX model is a round-trip time measured by the source host. Effects of other traffic (i.e., packets coming from other hosts) are modeled as the noise to the ARX model. Letting $u(k)$ and $y(k)$ be the input and the output at slot k, respectively, the ARX model is defined as

$$\begin{aligned} A(q)\, y(k) &= B(q)\, u(k - n_d) + e(k) \\ A(q) &= 1 + a_1 q^{-1} + \ldots + a_{n_a} q^{-n_a} \\ B(q) &= b_1 + b_2 q^{-1} + \ldots + b_{n_b} q^{-n_b+1} \end{aligned}$$

where $e(k)$ is unmeasurable disturbance (i.e., noise), and q^{-1} is the delay operator; i.e., $q^{-1}u(k) \equiv u(k-1)$. The numbers n_a and n_b are the orders of polynomials. The number n_d corresponds to delays from the input to the output. All coefficients of the polynomials, a_n and b_n, are parameters of the ARX model, and are to be identified from input and output data. Refer to [6] for the detail of the ARX model and system identification. For compact notation, ζ and θ are introduced as

$$\begin{aligned} \zeta &= [n_a, n_b, n_d] \\ \theta &= [a_1, \ldots, a_{n_a}, b_1, \ldots b_{n_b}]^T \end{aligned}$$

In [5], we have defined the input as the packet inter-departure time from the source host, and the output as the round-trip time variation measured by the source host. Although the ARX model, with such input and output definition, can capture the round-trip time dynamics to some extent, the model accuracy is not good. It is possibly because of the non-fixed sampling interval. Namely, use of a fixed sampling interval is generally assumed in system identification, however, in [5], the sampling interval is not fixed since it is dependent on the packet sending/receiving process at the source host. In this paper, we therefore use a fixed sampling interval for improving the model accuracy; that is, the input to the system is the packet transmission rate during a fixed sampling interval, and the output from the system is the average round-trip time during a fixed sampling interval. More specifically, the input $u(k)$ and the output $y(k)$ are defined as follows. Let $t_s(i)$ be the time at which the ith packet is injected into the network, and $t_r(i)$ be the time at which the ith ACK packet is received by the source host. We further introduce $l(i)$ as the size of the ith packet including the IP header, and T as the sampling interval. Then, $u(k)$ and $y(k)$ are defined as

$$\begin{aligned} u(k) &= \frac{\sum_{i \in \phi_s(k)} l(i)}{T} \\ y(k) &= \frac{\sum_{i \in \phi_r(k)} (t_r(i) - t_s(i))}{|\phi_r(k)|} \end{aligned}$$

where $\phi_s(k)$ (or $\phi_r(k)$) is the set of packet numbers sent (or received) during kth sampling interval; i.e.,

$$\begin{aligned} \phi_s(k) &\equiv \{n : kT \le t_s(n) < (k+1)T\} \\ \phi_r(k) &\equiv \{n : kT \le t_r(n) < (k+1)T\} \end{aligned}$$

3 Data Collection Using ICMP Packet

3.1 Measurement Method

For collecting input and output data from a real network, it is necessary to send a series of probe packets into the network, and to measure their resulting round-trip times. For sending a probe packet, one of the following protocols can be used.

- TCP (Transmission Control Protocol)
- UDP (User Datagram Protocol)
- ICMP (Internet Control Message Protocol)

In what follows, we briefly discuss advantages and disadvantages of these protocols for sending a probe packet to collect input and output data, in particular, for system identification.

TCP has a feedback-based congestion control mechanism, which controls the packet sending process from a source host according to the congestion status of the network. Since it is an ACK-based protocol, it is easy for the source host to measure the round-trip time for each packet. However, because of such a feedback-based mechanism, TCP is not suitable for sending a probe packet for two reasons. First, although the input (i.e., the packet transmission rate) should contain diverse frequencies for system identification purposes, the packet transmission rate of TCP would have limited frequencies. Second, regardless of many system identification techniques assuming an independence between the input and the output, the independence assumption cannot be satisfied with TCP since the packet transmission rate is dependent on the past round-trip times.

On the contrary, UDP has no feedback-based control. The packet transmission rate of UDP can be freely controlled. However, UDP is a one-way protocol. The destination host must perform some procedure to measure the round-trip time for each packet at the sender side. One possible way is to use *ICMP Destination Unreachable* message as in the *traceroute* program [7]. When the host receives a UDP packet to an unreachable port, it returns ICMP Destination Unreachable message to the source host. The source host can therefore measure the round-trip time by observing the elapsed time between the UDP packet transmission and the receipt of the corresponding ICMP packet. However, as specified in [8], generation of ICMP Destination Unreachable messages is limited to a low rate. Use of ICMP Destination Unreachable message is therefore not desirable to collect the input and output data for system identification.

ICMP is a protocol to exchange control messages such as routing information and node failures [9]. Since ICMP has no feedback-based control, the inter-departure time of ICMP packets can be freely controlled. Also it is easy to measure the round-trip time at the source host by using *ICMP Echo Request* and *ICMP Echo Reply* messages, as in the *ping* program. Although some network devices limit the rate of ICMP packets because of malicious use of them [10], such as a DoS (Denial of Service) attack, many network devices respond to ICMP Echo Request message and do not limit the rate of them.

In this paper, we therefore choose ICMP Echo message as a probe packet. More specifically, the source host sends a series of ICMP Echo Request messages to the destination host, and the destination host returns ICMP Echo Reply messages. We have modified the ping program to dynamically change the packet inter-departure time (originally fixed at one second). The destination host copies the payload of the received ICMP Echo Request message to the returning ICMP Echo Reply message. Thus, the ICMP Echo Reply packet contains the timestamp placed by the source host at its transmission time. This enables precise measurement of the round-trip time at the source host. Instead of measuring ICMP Echo Request/Reply packet sending/receiving time at the source, a *measurement host* is prepared (Fig. 3). It is for achieving reliable data measurement even when the source host sends or receives packets at a very high rate. As shown in Fig. 3, the *Ether TAP* copies all packets carried on the link, and sends copies to the measurement host; that is, all ICMP Echo Request/Reply packets sent from/to the source host are also delivered to the measurement host.

We use an active measurement approach for collecting data by sending probe packets to the network. This is because we want to know how accurately the ARX model can represent the round-trip time dynamics of the Internet. However, we intend to apply a passive measurement approach, which measures data by monitoring packets being transmitted in the network.

3.2 Network Environments

As the number of routers between source and destination hosts increases, the noise (i.e., effect of other traffic and measurement errors) contained in the output becomes large.

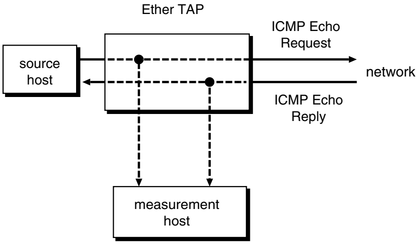


Fig. 3. Measurement host for reliable data measurement.

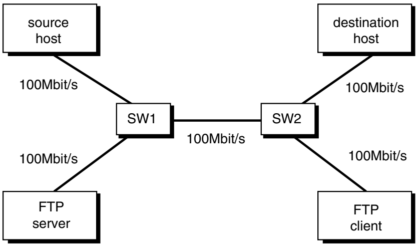


Fig. 4. Network **N1** (LAN)

Besides, the dominant part of the round-trip time is a queuing delay at the bottleneck router. It is therefore important to choose network configurations, in which the input and the output are collected, by taking account of the number of routers and the location of the bottleneck link. In this paper, we measure packet sending/receiving times in three network configurations including LAN and WAN environments, and obtain the input $u(k)$ and the output $y(k)$. In LAN environment, it is expected that the ARX model can accurately model the round-trip time dynamics since the network topology is rather simple and the measured data would suffer little observation noise. On the contrary, in WAN environment, it is expected that the model accuracy is degraded compared to that in LAN environment since the network topology is complex. We use two network configurations for WAN environment. The difference in these WAN configurations is the location of the bottleneck link. In this paper, the following three network configurations (i.e., **N1**, **N2**, and **N3**) are used for collecting input and output data.

- Network **N1** (LAN)

The network **N1** is LAN environment of a simple network configuration (Fig. 4). There exist two switches (SW1 and SW2) between source and destination hosts. All hosts and switches are connected to 100 Mbps LAN. The link between SW1 and SW2 also carries background traffic, as well as ICMP Echo Request/Replay packets exchanged between source and destination hosts. Namely, a bulk FTP transfer from a server (connected to SW1) to a client (connected to SW2) is performed during data collection.

- Network **N2** (WAN with the bottlenecked access link)

The network **N2** is WAN environment of a complex network configuration, and the access link is the bottleneck between source and destination hosts (Fig. 5). The source host is connected to the Internet via 100 Mbps LAN, and the destination host is connected via 56 Kbps dial-up PPP link. At the time of measurement, the number of hops between source and destination hosts was 16, and the average round-trip time was 319.7 ms.

- Network **N3** (WAN with the non-bottlenecked access link)

The network **N3** is WAN environment, and the access link is not the bottleneck between source and destination hosts (Fig. 6). The source host is connected to the Internet via 100 Mbps LAN. We have chosen `www.so-net.ne.jp` as the destination host. At the time of measurement, the number of hops between source and destination hosts was 16, and the average round-trip time was 36.89 ms.

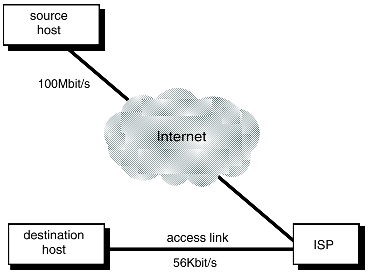


Fig. 5. Network **N2** (WAN with the bottlenecked access link)

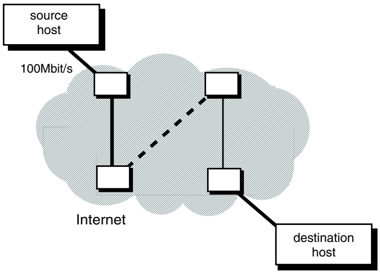


Fig. 6. Network **N3**: (WAN with the non-bottlenecked access link)

In the above three network configurations, we measured the packet sending/receiving time at the measurement host. The source host sent 20,000 ICMP Echo Request packets, and the timestamp of each ICMP Echo Request/Replay packet is recorded by the measurement host. The data collection was done at midnight of October 18, 2001. As we have explained in Section 2, the input $u(k)$ and the output $y(k)$ for system identification is calculated from measured packet sending/receiving times. We empirically choose the sampling interval T in each network configuration; that is, T is chosen for each sampling period to contain about five samples. In this paper, the packet inter-departure time from the source host is randomly changed, there might be a sampling period in which no packet is sent or received. If no packet is sent (or received) during kth sampling period, the input $u(k)$ (or the output $y(k)$) is not defined. In such a case, we use the minimum value of all past input (or output) data; i.e.,

$$
\begin{aligned}
u(k) &= \min_{0 \leq i < k} (u(k)) \\
y(k) &= \min_{0 \leq i < k} (y(k))
\end{aligned}
$$

4 Modeling from Measured Data

4.1 Choice of Model Orders and Number of Samples

In this paper, the orders of the ARX model are fixed at $n_a = 5$ and $n_b = 5$ in all three network configurations. This is for comparing the model accuracy in each network configuration. The delay from the input to the output, n_d, is determined from the average round-trip time. This is because the packet sending rate at a specific time would have influence on the packet receiving process after the round-trip time. By letting N be the total number of round-trip time samples, n_d is determined as

$$
n_d = \left\lfloor \frac{\sum_{k=1}^{N} y(k)}{N\,T} \right\rfloor \tag{1}
$$

For evaluating the model accuracy, we use two validation methods: (1) a validation method using simulation, and (2) a validation method in frequency domain. The first method is to compare the simulated output from the ARX model, where zero noise is assumed (i.e., $e(k) = 0$), with the actual output [6]. The simulated output from the ARX model is defined as

$$\begin{aligned} y^*(k|\theta) &= {\psi^*}^T(k|\theta)\,\theta \\ \psi^*(k|\theta) &= [-y^*(k-1|\theta), \ldots, -y^*(k-n_a|\theta), \\ &\qquad u(k-n_d-1), \ldots, u(k-n_d-n_b)] \end{aligned}$$

The ARX model is thought to be accurate if the simulated output from the ARX model coincides to the actual output.

The second method is to compare the frequency response of the ARX model with the frequency response estimated by the spectral analysis [6]. The bode plot is used for visually comparing those frequency responses. The bode plot illustrates the gain and the phase of a dynamic system at different frequencies. When a linear stable dynamic system has a sinusoid input signal

$$u(t) = A\sin(\omega\, t),$$

the output from the system can be written as

$$y(k) = B\sin(\omega + \phi)$$

The gain and the phase of the system at the frequency ω are B/A and ϕ, respectively. The frequency response of the ARX model is easily obtained by deriving its corresponding transfer function. On the contrary, the spectral analysis directly estimates the frequency response from measured input and output data. The ARX model is thought to be accurate if frequency responses of the ARX model and the spectral analysis are identical.

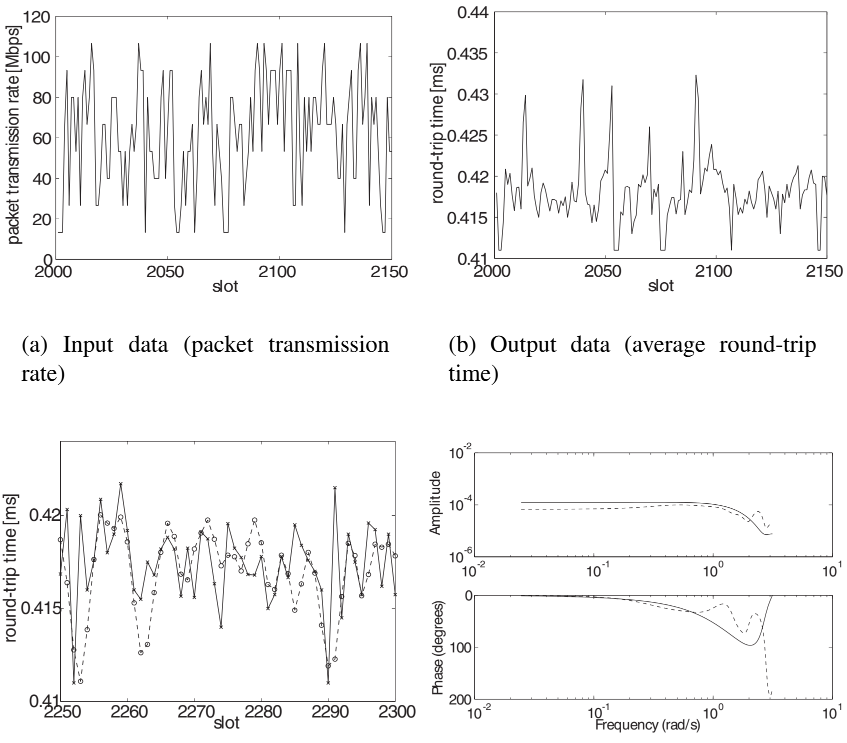
From all input and output data obtained in Section 3, two datasets for determining parameters of the ARX model (i.e., model identification) and for validating the model accuracy (i.e., model validation) are extracted. The number of input and output data used for system identification directly affects the model accuracy. We use 150 input and output data for both model identification and model validation; i.e., we use input and output data from 2,001 to 2,150 for model identification, and from 2,201 to 2,350 for model validation.

4.2 Modeling Results and Discussions

- Network **N1** (LAN)

Figure 7 shows (a) the input (the packet transmission rate from the source host), (b) the output (the average round-trip time), (c) comparison of the simulated output and the actual output, and (d) comparison of frequency responses of the ARX model and the spectral analysis for the network **N1**. Note that (a) and (b) are input and output data not for model validation, but for model identification. In this case, the average packet transmission rate from the source host was 66.6 Mbps, and the average round-trip time was 0.42 ms. The average throughput of the FTP transfer was 5.2 Mbps. Also the sampling interval is $T = 0.9$ ms, and the delay of the ARX model is $n_d = 0$ according to Eq. (1).

Figure 7(c) shows good agreement between the simulated output and the actual output. Figure 7(d) also shows good agreement between frequency responses of the



(a) Input data (packet transmission rate)

(b) Output data (average round-trip time)

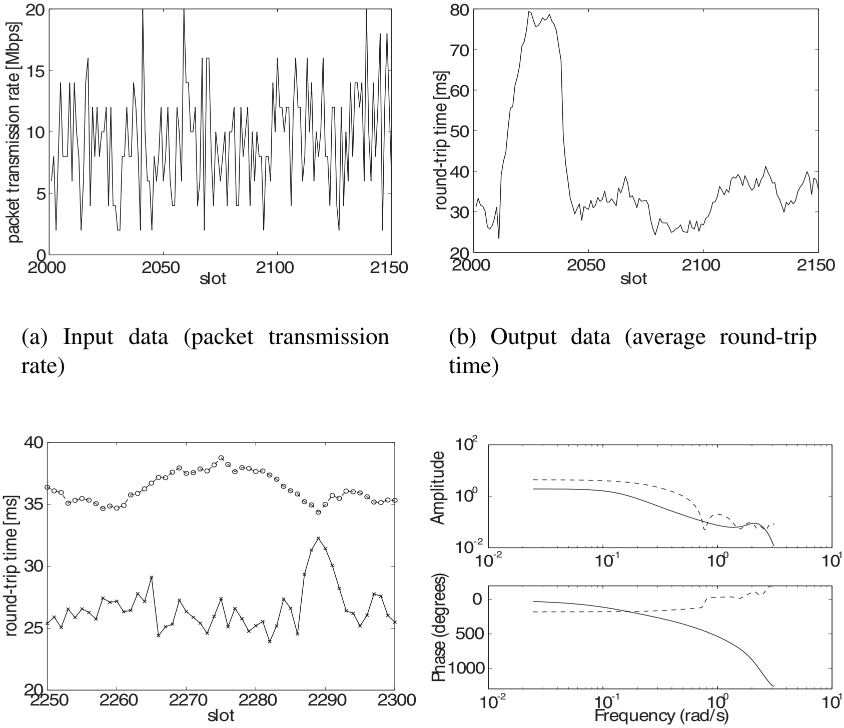
(c) Comparision with simulation (solid: measured output, dotted: simulated output)

(d) Comparision in frequency domain (solid: ARX model, dotted: spectrum analysis)

Fig. 7. Network **N1** (LAN)

ARX model and the spectral analysis. Although frequency responses at a high frequency are different, such disagreement would be caused by inaccuracy of the spectral analysis, in particular, at a high frequency [6]. From these observations, we conclude that in the network **N1**, the round-trip time dynamics can be accurately modeled by the ARX model. One of possible explanations for this phenomenon is that in LAN environment, the packet transmission rate from the source host directly affects the packet waiting time at the bottleneck link, resulting in a strong correlation between the packet transmission rate and the round-trip time.

Recall that the average packet transmission rate in this experiment is rather high (i.e., 66.6 Mbps). Although results are not included here due to space limitation, the ARX model cannot capture the round-trip time dynamics when the average packet transmission rate was 20 Mbps. This phenomenon is possibly because of little packet waiting time at the bottleneck link.



(a) Input data (packet transmission rate)

(b) Output data (average round-trip time)

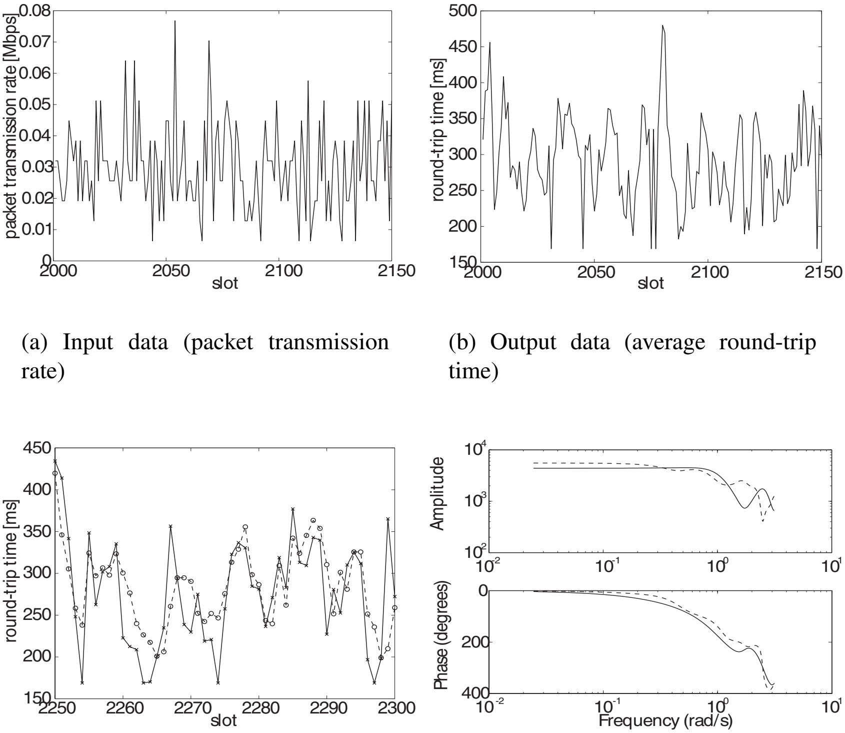
(c) Comparision with simulation (solid: measured output, dotted: simulated output)

(d) Comparision in frequency domain (solid: ARX model, dotted: spectrum analysis)

Fig. 8. Network **N3** (WAN with the non-bottlenecked access link)

- Network **N2** (WAN with the bottlenecked access link)

Figure 9 shows input and output data for model identification, and results of system identification for the network **N2**. In this experiment, the average packet transmission rate from the source host was 31.8 Kbps, and the average round-trip time was 319.7 ms. The sampling interval is $T = 125$ ms, and the delay of the ARX model is $n_d = 2$ according to Eq. (1). The network **N2** is WAN environment where there are 15 routers between source and destination hosts. Intuitively, it is expected that the modeling the round-trip time dynamics is more difficult than in the network **N1**. However, Fig. 9(c) indicates that the simulated output and the actual output well coincide. In addition, frequency responses in Fig. 9(d) shows good agreement between the ARX model and the spectral analysis. Hence, the round-trip time dynamics can be accurately modeled by the ARX model in the network **N2**. This phenomenon can be explained by the fact that the access link is the bottleneck between source and destination hosts. Namely, as the packet transmission rate



(a) Input data (packet transmission rate)

(b) Output data (average round-trip time)

(c) Comparision with simulation (solid: measured output, dotted: simulated output)

(d) Comparision in frequency domain (solid: ARX model, dotted: spectrum analysis)

Fig. 9. Network **N2** (WAN with the bottlenecked access link)

from the source changes, the packet waiting time also changes at the bottleneck link. In the network **N2**, since the access link is the bottleneck, the round-trip time suffers little disturbance from other traffic. Therefore, there exists a strong correlation between the packet transmission rate and the round-trip time, resulting in an accurate ARX model. From these observations and discussions, we conclude that in WAN environment, the round-trip time dynamics can be accurately modeled when the bottleneck link is shared by a small number of users.

- Network **N3** (WAN with the non-bottlenecked access link)

Figure 8 shows input and output data for model identification, and results of system identification for the network **N3**, where the average packet transmission rate from the source host was 9.55 Mbps, and the average round-trip time was 36.89 ms. The sampling interval is $T = 6$ ms, and the delay of the ARX model is $n_d = 6$. Figure 8(c) shows that the simulated output and the actual output are completely different. Also, frequency responses in Fig. 8(c) disagree. Thus, in the network **N3**, the ARX model fails to capture the round-trip time dynamics. This inaccuracy would be caused by a large noise; that is, in the network **N3**, the network topology is rather complex, and the bottleneck link would be shared by many users. Hence, the packet transmission rate from the source host has little impact on the round-trip time. In other words, there is a very weak correlation between the packet transmission rate and the round-trip time, resulting in an inaccurate ARX model. Although we have done several experiments for different destination hosts, the ARX model cannot capture the round-trip time dynamics in any case. From these observations, we conclude that the ARX model is unable to model the round-trip time dynamics when the network configuration is complex and the packet transmission rate has little effect on the round-trip time.

5 Conclusion

In this paper, we have modeled the round-trip time dynamics of the Internet by the ARX model using system identification. As input and output data for the ARX model, we have used the packet transmission rate from the source host and the average round-trip time measured by the source host. Using input and output data measured in working LAN and WAN environments, we have investigated how accurately the ARX model can capture the round-trip time dynamics. Through numerical examples, we have shown that in LAN environment, the round-trip time dynamics can be accurately modeled by the ARX model. We have also shown that in WAN environment, the round-trip time dynamics can be accurately modeled when the bottleneck link is shared by a small number of users.

As a future work, we are currently working to improve the model accuracy by using more complicated models such as ARMAX (Auto-Regressive Moving Average eXogenous) model. We are planning to design an efficient delay-based congestion control mechanism by utilizing the ARX model, which captures the round-trip time dynamics.

References

1. Internet Software Consortium, "Internet domain survey." available at `http://www.isc.org/ds/`.
2. R. Jain, "A delay-based approach for congestion avoidance in interconnected heterogeneous computer networks," *ACM Computer Communication Review*, vol. 19, pp. 56–71, Oct. 1989.
3. L. S. Brakmo, S. W. O'Malley, and L. L. Peterson, "TCP Vegas: New techniques for congestion detection and avoidance," in *Proceedings of ACM SIGCOMM '94*, pp. 24–35, Oct. 1994.
4. H. Ohsaki, M. Murata, and H. Miyahara, "Modeling end-to-end packet delay dynamics of the Internet using system identification," in *Proceedings of Seventeenth International Teletraffic Congress*, pp. 1027–1038, Dec. 2001.
5. H. Ohsaki, M. Morita, and M. Murata, "On modeling round-trip time dynamics of the Internet using system identification," to be presented at *the 16th International Conference on Information Networking (ICOIN-16)*, Jan. 2002.
6. L. Ljung, *System identification — theory for the user*. Englewood Cliffs, N.J.: Prentice Hall, 1987.
7. S. Hares, "Essential tools for the OSI Internet," *Request for Comments (RFC) 1574*, Feb. 1994.

8. F. Baker, “Requirements for IP version 4 routers,” *Request for Comments (RFC) 1812*, June 1995.
9. J. Postel, “Internet control message protocol,” *Request for Comments (RFC) 792*, Sept. 1981.
10. S. Savage, “Sting: A TCP-based network measurement tool,” in *USENIX Symposium on Internet Technologies and Systems*, pp. 71–79, Oct. 1999.

Optimal Link Capacity Dimensioning in Proportionally Fair Networks

Michał Pióro[1], Gábor Malicskó[2], and Gábor Fodor[3]

[1] Department of Communication Systems, Lund Institute of Technology, Sweden,
Michal.Pioro@telecom.lth.se
[2] Ericsson Traffic- and Performance Laboratory, Hungary,
Gabor.Malicsko@eth.ericsson.se
[3] Ericsson Research, Sweden, Gabor.Fodor@era-t.ericsson.se

Abstract. We consider the problem of link capacity dimensioning and bandwidth allocation in networks that support elastic flows and maintain *proportional fairness* among these flows. We assume that a certain allocated bandwidth to a user demand generates revenue for the network operator. On the other hand, the operator is incurred a capacity dependent cost for each link in the network. The operator's profit is the difference between the revenue and the total link cost. Under this assumption the problem is to determine the bandwidth of the flows and the link capacities such that the profit is maximized. We first show that under fairly general assumptions, the optimum allocation of flows leads to selecting the lowest cost paths between O-D pairs. We also derive explicit formulae for the bandwidth allocated to these flows. We distinguish the case when the operator's capacity budget is fixed ("equality budget constraint", in which case the profit is maximized when the revenue is maximized) and the case when the budget is upper-bounded ("inequality budget constraint", in which case the profit can - in general - be maximized by using some portion of the capacity budget). Finally, we show numerical examples to highlight some of the trade-offs between profit maximization, revenue maximization and fairness.

Keywords: network dimensioning, bandwidth allocation, routing, traffic engineering, linear programming, convex optimization

1 Introduction

After years of research and standardization efforts, there seems to be a growing consensus that some form of traffic engineering and in particular the separation of flows with different quality of service (QoS) demands are necessary to avoid too costly over-dimensioning of IP networks [1], [2]. To this end, MPLS provides a set of standards that can be applied to explicitly allocate bandwidth resources between originator-destination (O-D) pairs [13]. In addition, traffic engineering algorithms can also be useful to provide some kind of a "reference engineered network" that can help operators to determine the level of over-dimensioning in a non-engineered network. Despite these obvious motivations, it is however still the topic of research and standardization exactly which mechanisms and algorithms can be used in for instance MPLS networks for various aspects of traffic engineering.

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 277–288, 2002.

Despite numerous recent advances (see Section 2 for a more detailed discussion), adopting the "traditional" traffic engineering methods (including link capacity dimensioning, routing and bandwidth allocation) from circuit switched networks (such as PSTN's, ISDN's or even ATM networks) is non-trivial, because of the presence of elastic traffic classes.

Therefore, in this paper we concentrate on developing a model and algorithms that take into account the above three aspects of engineering for elastic services. This paper builds upon the results of the first author in [17] where the detailed proofs of the results are presented. Specifically, we assume that between each O-D pair, there may be only a single user flow *realizing* the demand, i.e. we exclude the case of the "demand split".

Under this assumption, the network operator faces the following problems:

- The capacity of each link must be determined such that the network can accommodate the offered traffic (dimensioning).
- The traffic demand between the O-D pair must be associated either with a single flow or - in the case when demand split is acceptable - with a set of flows, and an appropriate route must be found for each flow (routing).
- Bandwidth must be allocated for each flow such that some notion of fairness among the user flows is maintained (bandwidth allocation).

In our model, each user flow generates a bandwidth dependent *revenue* for the operator. On the other hand, each link incurs a capacity dependent *cost* for the operator, who is therefore motivated in maximizing the resource utilization in the network (and thereby its profit which is defined as the difference between revenue and cost). The network operator may be interested in maximizing its profit while keeping the total link cost fixed ("equality link cost budget constraint") or he may want to maximize the profit while keeping the total link cost under some bound ("inequality budget constraint"). Note that in the first case the profit is maximized when the revenue is maximized.

While there is no "killer argument" or any clear technical or economical evidence why a certain (if any) fairness between user flows should be maintained, maximizing the network throughput may lead to extremely unfair allocations (including the situation where some flows are deemed to complete starvation), see for instance [16].

Therefore, we will formulate the traffic engineering problem as a series of optimization tasks, where one is interested in maximizing the profit/revenue subject to capacity and fairness constraints. Specifically, we will consider the following two cases:

1. Dimensioning under a fixed budget constraint, the routes are considered fixed and pre-determined to each flow. (Task 1)
2. Profit (revenue-cost) maximization, where the budget (i.e. the sum of all link costs) is not fixed but can be freely chosen up to a limit. (Task 2)

Throughout the numerical evaluation of the techniques enumerated above we will use the proportional fair sharing with fixed link capacities and shortest path routing as a reference method (Task 3).

We organize the paper as follows. In the next section we take a look at recent research results in the area of bandwidth allocation in (max-min and proportional) fair networks. In Section 3 we formulate the problem of link capacity dimensioning and routing as a (revenue) optimization problem under budget and fairness constraints. We assume that the total link cost in the network should be equal to a pre-determined budget ("equality budget constraint"). For this case, we present explicit formulae to determine the allocated bandwidth for each flow. Next, we define the profit optimization task, in which case the sum of all link costs is bounded ("inequality budget constraint"). Here the task is to find both the link cost budget and the bandwidth allocated to flows (and associated revenue) such that the profit is maximized. In Section 6 we discuss numerical results. We conclude in Section 7.

2 Related Works

In the context of routing and resource allocation under fairness constraints, most paper consider the popular max-min fairness mostly in Asynchronous Transfer Mode (ATM) Available Bit Rate (ABR) context, since the ATM Forum adopted the max-min fairness criterion to allocate network bandwidth for ABR connections [7], [19], [21]. However, these papers do not consider the issue of path optimization in the bounded elastic environment. For instance, [21] studies the speed of convergence of max-min fair allocation algorithms rather than focusing on path or link capacity optimization.

From the point of view whether the flows are *static* (also called "long lived") or dynamic (where some arrival and departure patterns are also considered) these papers can be divided into two groups. Representative papers for the static case include for instance the papers by Kleinberg et al., see [11] and [12]. Another important series of work on the static case is the papers that develop on-line fair routing algorithms where the demand matrix is not a-priori known, see for instance [6], [18] and more recently [8].

The "dynamic case" under both max-min and proportional fairness is analyzed, mostly focusing on stability aspects, by Veciana *et al.* in [22]. Here, routes and link capacities are assumed fixed.

Max-min fair routing is the topic of the paper by Ma *et al.* [14], where the widest-shortest, shortest-widest and the shortest-dist algorithms are studied. These algorithms do not aim to explicitly maximize the carried traffic and consequently the path allocation is not formulated as an optimization task.

An important reference for both the static- and dynamic cases is the series of works by Massoulie and Roberts, see e.g. [15] and [16]. Here, a number of fairness notions are discussed and associated optimization tasks are presented for the case of the unbounded flows and assuming fixed routes and link capacities.

Although the max-min allocation has been widely accepted and studied in the literature, its appropriateness can be questioned because of the relatively low bandwidth utilization. One of the promising alternatives to the max-min fairness is the proportional-rate fairness proposed by Kelly in [9], [10] and also summarized by Massoulie and Roberts in [16]. Because of its superior characteristics in terms of overall network utilization, we in this paper concentrate on the proportional fair allocation method.

According to the proportional-rate fairness criterion, the rate allocations x_d are fair, if they maximize $\sum_d \log x_d$ (or in the weighted case $\sum_d w_d \log x_d$, where w_d is the weight of demand d) under the capacity constraints. This objective may be interpreted as being to maximize the overall revenue of allocations assuming each route has a logarithmic revenue function.

Bandwidth allocation algorithms can further be divided into two main groups. The ERAQLES algorithm in ATM [3] or the algorithms of [16] provide examples on distributed algorithms, while the application of a bandwidth broker facilitates the use of centralized bandwidth allocation algorithms [20].

In an earlier work [5], we developed centralized algorithms for networks with given (fixed) link capacities that optimize routing in order to improve the available level of fairness (either max-min or proportional rate) between flows. In [4] we combined network dimensioning and proportional fair bandwidth allocation into a single optimization task without the inequality budget constraint.

Our contribution to this line of works is the development of explicit analytical formulae for a set of optimization tasks that allow for the joint optimization of routing and link capacity dimensioning. To the best of our knowledge, such a profit/revenue optimization formulation and associated formulae and algorithms have not yet been proposed in the literature.

3 Fixed Budget Network Dimensioning for Non-bounded Elastic Flows

As a basic case, consider the following optimization task, where the objective is to find the rate allocations x_d and link capacities y_e such that the logarithmic revenue function corresponding to proportional fairness is maximized. Note that each link e is associated with a *marginal cost* c_e and so the operator's total link cost is $\sum c_e y_e$. In this basic setting we assume that the operator's total link cost budget (C) is kept fixed such that $C = \sum c_e y_e$.

Task 1. Dimensioning under Fixed (Equality) Budget Constraint with Fixed Routes

indices:

$d = 1, 2, ..., D$ demands
$e = 1, 2, ..., E$ links

constants:

$$a_{ed} = \begin{cases} 1 \text{ if link } e \text{ belongs to the path realizing demand } d \\ 0 \text{ otherwise.} \end{cases}$$

w_d weight of demand d
c_e marginal cost of link e
C assumed budget

variables:

x_d flow allocated to demand d
y_e capacity of link e

maximize:

$$F(x) = \sum_d w_d log(x_d) \tag{1}$$

constraints:

$$\sum_e c_e y_e \qquad = \qquad C; \tag{2}$$

$$\sum_d a_{ed} x_d \qquad \leq \qquad y_e; \quad e = 1, 2, ..., E \tag{3}$$

x,y non-negative.

The explicit solution of the optimization task is given by the following theorem. **Theorem 1.** Let x^0 and y^0 be the solution of the above task. Then:

$$F(x)|_{x=x^0} = log(C) \sum_d w_d - \sum_d w_d log(\sum_e c_e a_{ed}) +$$

$$+ \sum_d w_d log(w_d) - log(\sum_d w_d) \sum_d w_d \tag{4}$$

$$x_d^0 = C w_d / ((\sum_d w_d)(\sum_e c_e a_{ed})) \tag{5}$$

$$y_e^0 = \sum_d a_{ed} x_d^0. \tag{6}$$

Proof: As shown in [17], the dual function for Task 1 is of the form (7) where σ is a non-negative dual variable (multiplier) corresponding to constraint (2).

$$W(\sigma) = \sum_d (w_d - w_d log(w_d / \sigma \sum_e c_e a_{ed})) - \sigma C. \tag{7}$$

The maximum of the dual function is attained at the stationary point of (7) with respect to the multiplier :

$$\sum_d w_d / \sigma - C = 0. \tag{8}$$

Hence, the optimal multiplier σ^0 is given by

$$\sigma^0 = \sum_d w_d / C \tag{9}$$

and this immediately implies (4) and (5). ■

Naturally, at the optimum the constraint of inequality (3) is binding. We note that the maximum value of the objective function (1) depends only on C; (1) implies that this maximum is of the form

$$F(C) = \alpha log(C) + \beta - \gamma. \tag{10}$$

where $\alpha = \sum_d w_d$, $\beta = \sum_d w_d log(w_d) - log(\sum_d w_d) \sum_d w_d$ and $\gamma = \sum_d w_d log(\sum_e c_e a_{ed})$.

Formula (10) implies that when the paths for the flows realizing the demands and the link capacities are also subject to optimization then the optimal solution will assign each flow to its shortest (lowest cost with respect to the c_e-s) path . This is because in (10) only γ depends on the path selection and it is minimized when the lowest cost paths are used for realizing the demands' flows.

4 Profit Maximization

During the dimensioning of a network the budget constraint usually appears only as an upper limit on the disposable amount of money and the target is to achieve an investment that maximizes the difference of the revenue and the total link costs ("profit"). The revenue associated with demand d depends on the operator's charging model and is not necessarily a linear function of the allocated bandwidth (x_d). In fact, a logarithmic revenue function can be considered appropriate [23]. This logarithmic function is also motivated by the observation that the revenue can become negative if the allocated bandwidth is smaller than a threshold value. The optimization task in accordance with this objective is the following:

Task 2. Profit Maximization under Flexible (Inequality) Budget Constraint with Fixed Routes
maximize:

$$\Psi = \sum_d w_d log(x_d) - \sum_e c_e y_e \tag{11}$$

constraints:

$$\sum_e c_e y_e \leq C_0; \tag{12}$$

$$\sum_d a_{ed} x_d \leq y_e; e = 1, 2, ..., E \tag{13}$$

x,y non-negative.

The maximal value of the objective function (11) is attained at the maximum, with respect to variable C, of the function

$$F(C) = log(C) \sum_d w_d - C \tag{14}$$

$$\text{over } 0 \leq C \leq C_0. \tag{15}$$

The optimum of (14) is attained either at $C = \sum_d w_d$ (if $\sum_d w_d \leq C_0$) or at C_0 (if $\sum_d w_d > C_0$). Of course, the optimal C is the optimal total cost of links $\sum_e c_e y_e$. Furthermore, from (4) it follows that in the case of $\sum_d w_d \leq C_0$ the optimal value of (11) is equal to

$$\sum_d w_d log(w_d/\xi_d) - \sum_d w_d \tag{16}$$

(where ξ_d is the length of the shortest path of demand d), and the optimal flows are given by

$$x_d^0 = w_d/\xi_d; \qquad d = 1, 2, ..., D. \tag{17}$$

5 The Fixed Link Capacity Method

We are interested in analyzing the differences between the method outlined in Task 1 and Task 2 and the proportional fair allocation mechanism used for fixed link capacities. In order to do this we need to formulate explicitly the allocation task used for comparisons. The numerical results are presented in Section 6.

Task 3. Proportional Fair Allocations for Links of Fixed Capacity and Lowest Cost Path Routing

indices:

$d = 1, 2, ..., D$ demands

$e = 1, 2, ..., E$ links

constants:

$$a_{ed} = \begin{cases} 1 \text{ if link } e \text{ belongs to the path realizing demand } d \\ 0 \text{ otherwise.} \end{cases}$$

w_d weight of demand d

c_e marginal cost of link e

C assumed budget

$y_e = \frac{C}{\sum_e c_e} c_e$ capacity of link e

variables:

x_d flow allocated to demand d

maximize:

$$F(x) = \sum_d w_d log(x_d) \tag{18}$$

constraints:

$$\sum_d a_{ed} x_d \leq y_e; \tag{19}$$

$$\text{for} \quad e = 1, 2, ..., E$$
$$d = 1, 2, \ldots, D$$

In Task 3, between each O-D pair we choose the lowest cost path with respect to the links' marginal cost c_e (lowest cost path routing). Note that the formulation allows for links of different capacities, but during the numerical evaluation we will consider equal sized links (where $c_e \equiv 1$). There is no closed formula available for the calculation of the allocations in this case, therefore we used the optimization tool "Solver" included in Microsoft Excel for numerical evaluations. We used the piece-wise approximation of the logarithmic revenue function as described in [5].

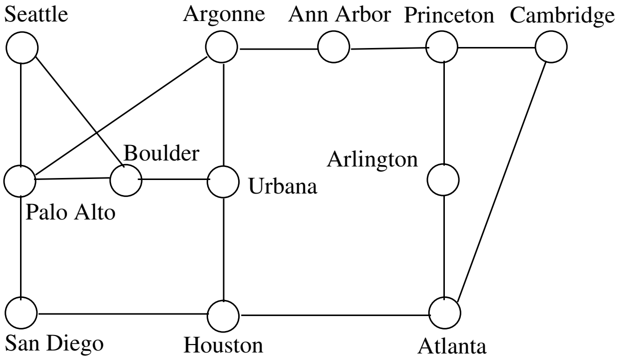


Fig. 1. The 12-node Example Network

6 Numerical Examples

6.1 Input Parameters

We consider the network of Figure 1. This network consists of 12 nodes and $E = 16$ links. We assume that there is one demand between each network node pair yielding in total $D = 66$ demands. We assume that each link has unit marginal cost (i.e. $c_e \equiv 1 \ \forall e$) and that each demand has equal weight ($w_d \equiv 10 \ \forall d$).

6.2 Numerical Results

Recall from the formulation of Task 1 and Task 2 that the revenue is associated with the allocated bandwidth only, while the profit takes into account the link costs as well. Figure 2 compares the optimum revenue and profit as the function of the link cost budget C of Task 1 ("Revenue" and "Profit") with those of Task 2 ("Max-Profit" and "Max-Revenue"). Recall that for a fixed budget C, Task 1 determines the bandwidth of each flow (x_d) and the capacity of each link (y_e) such that the revenue is maximized (see (1)). Since Task 1 uses the equality budget constraint, in this case the profit is maximized when the revenue is maximized and the difference between these two quantities is exactly C. (Indeed, see the "Revenue" and "Profit" curves.) For Task 2, x_d, y_e and the actually used link budget are determined such that the profit is maximized (see (11)). (That is, for Task 2, the horizontal axis corresponds to the maximum allowable budget C_0 of (12).) Figure 2 plausibly highlights the difference between the revenue and profit maximization tasks. Increasing the budget up to a certain limit leads to the increase of both the "Profit" and "Max-Profit". Increasing the budget beyond this point (in this example at $C = 660$) decreases the "Profit" of Task 1 and leads to the saturation of "Max-Profit" of Task 2, since in the profit maximization case the actually used budget may be smaller than the allowable. Task 2 effectively "freezes" the revenue increase that would lead to profit decrease (see the curve "Max-Revenue").

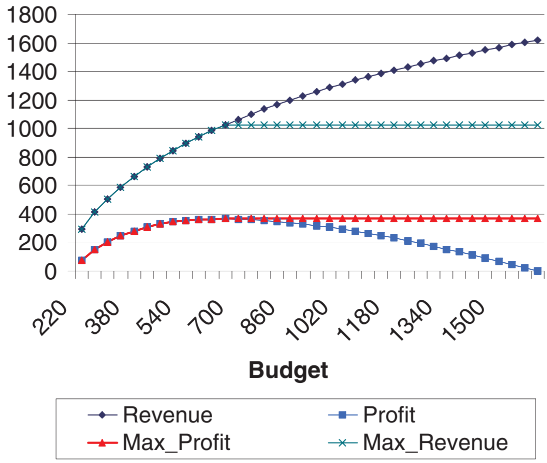


Fig. 2. Revenue and Profit for Task 1 and Task 3

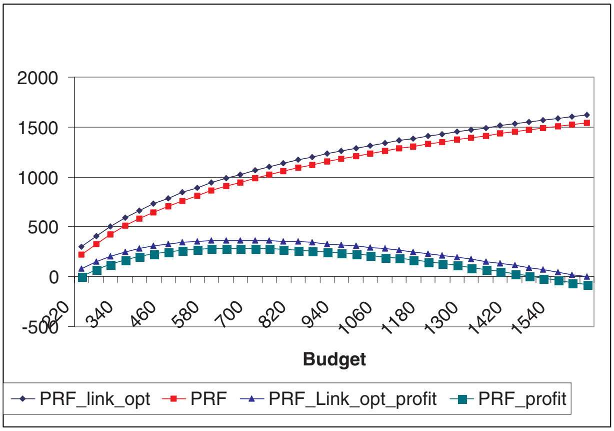


Fig. 3. Revenue and profit for fixed and optimized link PRF

In Figure 3 we compare the revenue and the profit as the function of the link cost budget C of Task 1 ("PRF_link_opt" and"PRF_link_opt_profit") with those provided by the fixed link proportional fair method of Task 3 ("PRF" and "PRF_profit"). We can see again analogously to Figure 2 the logarithmic increase of the revenue in case of both methods and the saturation of the profit function around the same link cost budget value of $C = 660$. Note that both the revenue and consequently the profit values are significantly higher when the link capacities are optimized (Task 1). Moreover, the difference between the two methods is approximately constant for both aspects.

The differences are even more visible in Figure 4 that highlights the difference between the revenue ("Revenue") and the profit ("Profit") values of Task 1 and Task

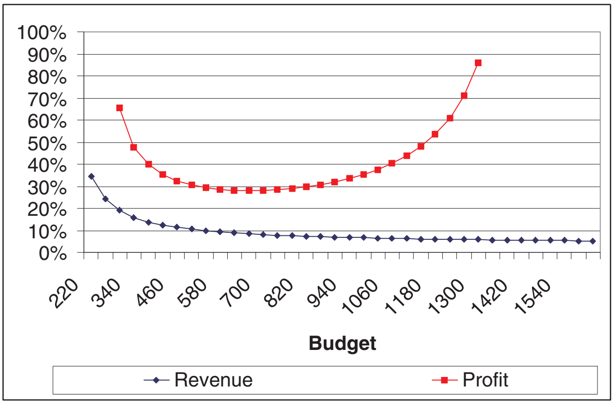


Fig. 4. Revenue and profit for fixed and optimized link PRF II.

3. The vertical axis is the "gain of Task 1 as compared to Task 3": it is calculated as the profit (and revenue) given by the optimized link proportional fair method (Task 1) divided by that of the fixed link case (Task 3), minus 1.

In accordance with the approximately constant difference between the revenues provided by Task 1 ("PRF_link_opt") and Task 3 ("PRF") outlined in Figure 3, the relative gain provided by Task 1 is monotonously decreasing. We can observe in Figure 3, that at low capacity budget values, the profit of both Task 1 ("PRF_link_opt_profit") and Task 3 ("PRF_profit") are low due to the insufficient budget. Thus, the relative gain in Figure 4 when optimizing the capacity (Task 1) as compared to the fixed capacity case (Task 3) is high. As the capacity budget increases, the reachable optimum profit grows, and as the difference between the Task 1 and Task 3 is more or less constant, the relative profit gain decreases, arriving to its minimum at the optimal capacity budget. Note, that even in this case link capacity dimensioning adds (in our example approximately 30%) to the profit as compared to the fixed capacity case. When the budget goes beyond the optimum, the profit decreases again and the relative gain of Task 1 as compared to Task 3 becomes higher again.

7 Conclusion

In this paper we considered link capacity dimensioning and bandwidth allocation in proportionally fair networks. We first considered the "basic task" (Task 1) where one is interested in finding the flow bandwidths and link capacities such that the revenue is maximized. For this case we provided explicit exact formulae. An important variant of this type of optimization problem arises when the cost associated with the total link capacities is not kept fixed but upper-bounded. In this case (Task 2) the profit (defined as the difference between revenue and cost) is maximized by using a portion of the total allowed link capacity budget. As a "reference case" for comparisons, we also considered the proportional fair sharing method with fixed link capacities (Task 3).

We summarize our findings as follows:

- Under the fixed budget constraint, when both the paths for the flows realizing the demands and the link capacities are subject to optimization, the optimal solution assigns each flow to its shortest (with respect to the links' marginal cost) path.
- Under the inequality budget constraint, i.e. when the actually used link capacity budget can be chosen up to a bound, the maximum profit can be reached at a significantly lower capacity budget value than the maximum allowed budget.
- Link capacity dimensioning in proportionally fair networks may significantly increase the profit as compared to the case when the link capacities are fixed.

We believe that our problem formulations in Task 1 and Task 2 can provide important insight into traffic engineering problems and can serve as a basis for practically useful engineering tools. In future work we plan to present results for the case when the user flows are both lower- and upper bounded as the model in [17].

References

1. G. R. Ash, "Traffic Engineering & QoS Methods for IP-, ATM-, & TDM-Based Multiservice Networks", draft-ietf-tewg-qos-routing-00.txt, http://www.ietf.org/internet-drafts/draft-ietf-tewg-qos-routing-00.txt, *Internet Engineering Task Force*, work in progress, November 2000.
2. D. O. Awduche, A. Chiu, A. Elwalid, I. Widjaja, Xipeng Xiao, "A Framework for Internet Traffic Engineering" draft-ietf-tewg-framework-02.txt, http://www.ietf.org/internet-drafts/draft-ietf-tewg-framework-02.txt, *Internet Engineering Task Force*, work in progress, July 2000.
3. A. Fichou, C. Galand, S. Fdida, Y. Moret, "Evaluation of the ER Algorithm ERAQLES in Different ABR Environments", 5^{th} *IFIP TC6 Workshop on Performance Modeling and Evaluation of ATM Networks*, pp. 48/1-48/3, Ilkley, UK, July 1997.
4. G. Fodor, G. Malicsko, M. Pioro, "Link Capacity Dimensioning and Path Optimization for Networks Supporting elastic Services", submitted to *ICC 2002*, New York, USA, 2002.
5. G. Fodor, G. Malicsko, M. Pioro and T. Szymanski, "Path Optimization for Elastic Traffic under Fairness Constraints", 17^{th} *International Teletraffic Congress*, Salvador da Bahia, Brasil, 2001.
6. A. Goel, A. Meyerson, S. Plotkin, "Combining Fairness with Throughput: Online Routing with Multiple Objectives", To appear in JCSS, special issue on Internet Algorithms (Invited Paper) Extended abstract in STOC 2000
7. Y. T. Hou, H. H-Y. Tzeng, S. S. Panwar, "A Generic Weight-Based Network Bandwidth Sharing Policy for ATM ABR Service", *IEEE International Conference on Communications, ICC '98*, pp. 1492-1499, 1998.
8. K. Kar, M. Kodialam, T. V. Lakshman, "Minimum Interference Routing of Bandwidth Guaranteed Tunnels with MPLS Traffic Engineering Applications", *IEEE Journal on Selected Areas in Communications*, Vol. 18, No. 12, pp. 2566-2579, Dec. 2000.
9. F. P. Kelly, A. K. Mauloo, D. K. H. Tan, "Rate Control for Communication Networks: Shadow Prices, Proportional Fairness and Stability", *Journal of the Operational Research Society*, (49), pp. 206-217, August, 1997.
10. F. P. Kelly, A. K. Maulloo, D. K. H. Tan, "Rate Control for Communication Networks: Shadow Prices, Proportional Fairness and Stability", *Journal of the Operational Research Society*, (49), pp. 237-252, 1998.

11. J. Kleinberg, "Single-Source Unsplittable Flow", *IEEE Symposium on Foundations of Computer Science*, 1996.
12. J. Kleinberg, Y. Rabani, É. Tardos, "Fairness in Routing and Load Balancing", *IEEE Symposium on Foundations of Computer Science*, 1999.
13. J. Lawrence, "Designing Multiprotocol Label Switching Networks", *IEEE Communication Magazine*, pp. 134-142, Vol. 39, No. 7, July 2001.
14. Q. Ma, P. Steenkiste, H. Zhang, "Routing High-bandwidth Traffic in Max-min Fair Share Networks", *SIGCOMM '97*, pp. 206-217, August, 1997.
15. L. Massoulie, J. W. Roberts, "Bandwidth Sharing and Admission Control for Elastic Traffic", *International Teletraffic Specialist Seminar*, Yokohama, 1998.
16. L. Massoulie, J. W. Roberts, "Bandwidth Sharing: Objectives and Algorithms", *IEEE INFOCOM '99*, 1999.
17. M. Pioro, "On Some Dimensioning Tasks Associated with the Notion of Proportional Fairness", *Technical Report, Lund Institute of Technology at Lund University*, CODEN:LUTEDX(TETS-7181)/1-6/(2001)&local 18, October 2001. http://www.telecom.lth.se/Personal/Michal.Pioro/report.pdf
18. S. Plotkin, "Competitive Routing of Virtual Circuits in ATM Networks", *IEEE Journal of Selected Areas in Communications*, Vol. 13, No. 6, pp.1128-36, Aug. 1995.
19. H. Qingyanga, D.W. Petr, "Global Max-Min Fairness Guarante for ABR Flow Control", *IEEE INFOCOM '98*, 1998.
20. B. Teitelbaum, S. Hares, L. Dunn, R. Neilson, V. Narayan, F. Reichmeyer, "Internet2 QBone: Building a Testbed for Differentiated Services", *IEEE Network*, Vol. 13, No. 5, pp. 8-16, September/October 1999.
21. W. K. Tsai, M. Iyer, "Constraint Precedence in Max-Min Fair Rate Allocation", *IEEE International Conference on Communications, ICC*, 2000.
22. G. de Veciana, T-J. Lee, T. Konstantopoulos, "Stability and Performance Analysis of Networks Supporting Elastic Services", *IEEE/ACM Transactions on Networking*, Vol. 9, No. 1, pp. 2-14, Feb. 2001.
23. Andreu Mas-Collel, Michael D. Whinston, Jerry R. Green, "Microeconomic Theory" *New York, Oxford University Press* 1995.

Load Balancing in WDM Networks through Adaptive Routing Table Changes

Mauro Brunato[⋆], Roberto Battiti, and Elio Salvadori

Università di Trento, Dipartimento di Informatica e Telecomunicazioni
via Sommarive 14, I-38050 Pantè di Povo (TN), Italy
battiti|brunato|salvador@science.unitn.it

Abstract. In this paper we develop a Load Balancing algorithm for IP-based Optical Networks. The considered networks are based on a routing protocol where the next hop at a given node depends only on the destination of the communication. Our algorithm (RSNE - Reverse Subtree Neighborhood Exploration) performs at each iteration a basic change of a single entry in a routing table in order to minimize the disruption of the network.
We study the performance of our algorithm in realistic networks under static and dynamic traffic scenarios. Simulation results show a rapid reduction of the congestion for static networks and a performance of the incremental scheme while tracking a changing traffic matrix comparable to the complete reoptimization of the traffic.

Keywords: WDM, load balancing, local search, dynamic traffic.

1 Introduction

Wavelength Division Multiplexing (WDM) and Generalized Multi Protocol Label Switching (G-MPLS) have been proposed to support the growing bandwidth demand caused by the exponential Internet growth and to permit suitable traffic engineering. In WDM networks, a wavelength is assigned to each connection in such a way that all traffic is handled in the optical domain, without any electrical processing on transmission [1]. The established lightpaths form the virtual or logical topology, opposed to the network physical topology composed of nodes (Optical Cross-Connects - OXCs) and fibers.

Current advances in optical communication technology are rapidly leading to flexible, highly configurable optical networks. The near future will see a migration from the current static wavelength-based control and operation to more dynamic IP-oriented routing and resource management schemes. Future optical networks designs should probably be based on fast circuit switching, in which end-to-end optical pipes are dynamically created and torn down by means of signaling protocols and fast resource allocation algorithms [5].

IP modifications are being proposed to take QoS requirements into account and to integrate the IP protocol within the optical layer. At the same time, a generalized version of Multi-protocol Label Switching (G-MPLS) is currently being developed to enable

[⋆] Corresponding Author.

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 289–300, 2002.

fast switching of various type of connections, including lightpaths. As soon as protocol modifications can ensure different QoS levels at the IP level, more and more statically allocated traffic can be transmitted on the dynamic portion of the network leading to an all optical and fully dynamic G-MPLS controlled optical cloud [12]. In this scenario it is necessary to study the impact of routing mechanisms typical to the IP world.

The basic motivation behind Load Balancing in computer networks is to reduce the congestion in the network. Congestion is related to delays in packet switching networks, and therefore reducing congestion implies better quality of service guarantees. In networks based on circuit switching (see for example the G-MLPS protocol), reducing congestion implies that a certain number of spare wavelengths are available on every link to accommodate future connection requests or to maintain the capability to react to faults in restoration schemes. In addition, reducing congestion means reducing the maximum traffic load on the electronic routers connected to the fibers.

Load balancing leads to the problem of creating virtual connections by considering both routing and wavelength assignment. The routing problem has its origin at the beginning of networking research (see [9] for a review of previous approaches to the problem). In particular adaptive routing, which incorporates network state information into the routing decision, is considered in [8] in the context of all-optical networks, while previous work on state-dependent routing with trunk reservation in traditional telecommunications networks is considered in [7]. It is also known that flow deviation methods [2], although computationally demanding, can be used to find the optimal routing that minimizes the maximum link load for a given network topology.

Because global changes of the logical topology and/or routing scheme can be disruptive to the network, we consider algorithms that are based on a sequence of small steps (i.e., on local search from a given configuration). In [4] "branch exchange" sequences are considered in order to reach an optimal logical configuration in small steps, upper and lower bounds for minimum congestion routing are studied in [13], where variable depth local search and simulated annealing strategies are also proposed. Strategies based on small changes at regular intervals are proposed in [9].

Our technological context is that of dynamic lightpath establishment in wavelength-routed networks reviewed in [14]. We therefore assume a mechanism to assign resources to connection requests, that must be able to select routes, assign wavelengths and configure the appropriate logical switches, see also [3] for integrated IP and wavelength routing and [6] for a blocking analysis in the context of *destination initiated reservation*.

This paper describes a preliminary investigation on protocols that consider IP-like routing strategies, where the next hop at a given node is decided only by the destination of the communication. In particular, we consider a basic change in the network that affects a single entry in the routing table of one node. In the context of all-optical networks this is relevant for optical packet switching networks, or for circuit switching networks (e.g. based on G-MPLS) where the optical cross-connects allow arbitrary wavelength conversion. The focus of this work is to study basic mechanisms in a simplified context. We plan to extend the work in the future by considering more general routing mechanisms (label switched paths in G-MPLS) and limited or no wavelength conversion.

Let us introduce the terminology that shall be used throughout this work. A *routing table* is an array, associated to each node of the network, containing next-hop information required for routing. The *traffic pattern* is available as an $N \times N$ matrix (N being the number of nodes in the network) $T = (t_{ij})$ where t_{ij} denotes the number of lightpaths (or the number of traffic load units) required from node i to node j. We assume that the entries t_{ij} are non-negative integers and $t_{ii} = 0$ for all i. Given a traffic pattern and a routing table on each node, the sum of the number of lightpaths passing through each link is called the *virtual load* of the link. Finally, the maximum virtual load along a path is called the *congestion* of the path. The maximum virtual load on the whole network is called the *congestion* of the network.

The Load Balancing problem is defined as follows.

> LOAD BALANCING — Given a physical network with the link costs and the traffic requirements between every source-destination pair (number of lightpaths required), find a routing of the lightpaths for the network with least congestion.

In the following sections, first we introduce the Reverse Subtree Neighborhood Exploration (RSNE) algorithm in Sect. 2 and then discuss the implementation of an incremental version (I-RSNE) in Sect. 3. Finally Sect. 4 analyses simulation results, by considering both the static and the dynamic traffic cases.

2 Local Search for the Load Balancing Problem

In this paper we propose a new scheme based on a simple Local Search heuristic, the *Reverse Subtree Neighborhood Exploration* (RSNE). The basic idea behind this scheme is the following: start by setting a shortest path routing, then — iteratively — try to minimize the congestion of the network by rerouting part of the traffic passing through the most congested link in the network. Rerouting is not necessarily performed at the ingress node of the congested link, as all nodes lying on routes that pass through the congested link (the *upstream nodes*) shall be considered by the algorithm for a possible change of their routing tables.

Refer to Fig. 1 for the following explanation. Consider the simplified hypothesis of a network with a unique most congested link as depicted in the upper part of the figure: we can identify the congested link with its endpoints (*cFrom*, *cTo*). In this special case there are six lightpaths crossing that link, three of them coming from node src_a, one coming from src_b and two from src_c. Three lightpaths are directed to destination node $dest_1$, all others to $dest_2$.

A first approach to reduce the load on the congested link is to consider one of the destination nodes (e.g. $dest_2$) and reroute part of the load addressed to it from the congested link to some other neighbor nb_i of $cFrom$, provided that the new route does not end up in a cycle and that the congested link is avoided. This move is achieved by modifying only one single entry of the $cFrom$ routing table (see it on the upper right side of Fig. 1), e.g. from cTo to nb_2. In this example, three lightpaths are removed from the congested link $(cFrom, cTo)$ and are rerouted through the link $(cFrom, nb_2)$. All destinations and all neighbors of $cFrom$ are considered before choosing the actual routing table entry to change and its new value. This allows to choose the best option. Actually, we found (as pointed out in Sect. 4) that even if the best possible move increases the congestion there is still reason to choose it, because further improvement could arise

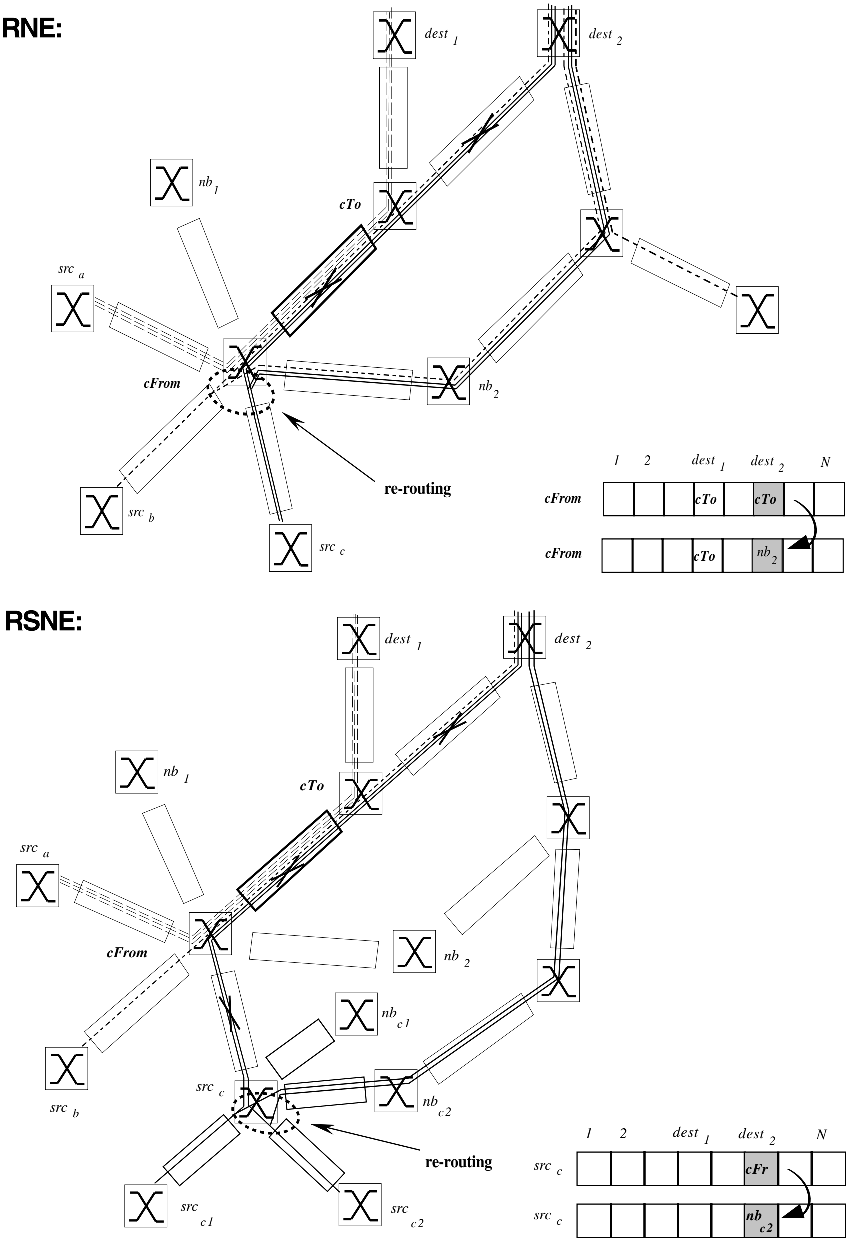


Fig. 1. *Restricted Neighborhood Exploration* (RNE) and *Reverse Subtree Neighborhood Exploration* (RSNE).

in the following steps. The algorithm stops when a predetermined number of iterations has been performed, or when all possible moves end up with a nonconsistent routing table (one causing loops or disconnected node pairs). The approach just described is called *Reduced Neighborhood Exploration* (`RNE`); we call it *reduced* to put it in contrast with the following extension.

Consider now the lower part of Fig. 1, which reproduces a larger portion of the same graph. To reroute part of the load addressed to, e.g., $dest_2$ and crossing the congested link $(cFrom, cTo)$ the search may be extended to all upstream nodes whose routes to $dest_2$ cross the congested link. The routing is destination-driven, therefore one can always identify the tree composed of all the links lying on lightpaths to a specified destination $dest_i$. It is straightforward to get the subtree rooted in $cFrom$ and composed of all the links lying on lightpaths to node $dest_2$: in Fig. 1 it is identified by nodes src_b, src_c, src_{c1}, src_{c2}. In this case, taking into consideration one of the nodes composing this subtree (e.g. src_c), we could try to reroute part of the load on the most congested link towards some of the downstream src_c's neighbors nodes nb_{ci}, while avoiding cycles and the use of the congested link. This local move is realized again modifying one single entry of the node's routing table (see it on the lower right side of Fig. 1, e.g. from $cFrom$ to nb_{c2}). In this case, only two lightpaths are removed from $(cFrom, cTo)$ by sending them through an alternate path to $dest_2$. Even though the improvement is smaller than in the previous case, where only the neighbors of $cFrom$ were considered, we shall see in Sect. 4 that, by allowing such fine-grain variations, this more general scheme achieves much better results. Again, all possible moves are considered before choosing a routing table change. This implies scanning all possible destination nodes having $(cFrom, cTo)$ in their routing tree and, for each destination node, all neighbors of every upstream node of $cFrom$. Even congestion increases are accepted, if no improving option is found. This technique is called *Reverse Subtree Neighborhood Exploration* (`RSNE`).

Fig. 2 shows an outline of our Local Search algorithm used for the Load Balancing problem: the initialization section (lines 1–2) starts by generating the routing tables through the application of the Shortest Path Routing algorithm to the specific network. By using the function *calculateLoad* we initially calculate the load on each link of the network, the initial value of *congestion* (from which the local search algorithm starts its search of the minimum) and the set of congested links.

The rest of the algorithm is a loop (lines 3-23) containing the local search algorithm.

The functions, variables and data structures used throughout this block have the following meaning:

- The set *candidateMoveSet* contains all candidate routing table changes. Its elements are triplets whose components are the node whose table must be changed, the index of the entry and the value that replaces the one already present.
- The function *routingTree*(*d*,*r*) returns the subtree that contains the nodes whose communications directed to destination *d* pass through node *r*.
- The function *shortestPathRouting*(*network*) calculates the shortest path tree for each destination node and returns the corresponding routing table as a matrix.
- The vector *rTable*[*n*] is the routing table of node *n*, whose *i*-th entry *rTable*[*n*][*i*] is the next-hop node index for lightpaths passing through node *n* and with destination *i*.

```
1.  rTable ← shortestPathRouting(network)
2.  <congestion,congestedLinkSet> ← calculateLoad(network,traffic,rTable)
3.  repeat
4.      bestCandidateLoad ← $+\infty$
5.      candidateMoveSet ← $\emptyset$
6.      for each link <cFrom,cTo> ∈ congestedLinkSet
7.          for each destination node dest such that rTable[cFrom][dest]=cTo
8.              for each node src ∈ routingTree(dest,cFrom)
9.                  removePartialLoad (src, dest)
10.                 for each neighbor node nb ∈ neighborhood(src)
11.                     vl ← virtual load on the candidate path from nb to dest
12.                     if (vl = bestCandidateLoad)
13.                         candidateMoveSet ← candidateMoveSet ∪ {<src,dest,nb>}
14.                     else if (vl < bestCandidateLoad)
15.                         bestCandidateLoad ← vl
16.                         candidateMoveSet ← {<src,dest,nb>}
17.                 restorePartialLoad (src, dest)
18.     if ( candidateMoveSet $\neq \emptyset$ )
19.         <src,dest,nb> ← pickRandomElement ( candidateMoveSet )
20.         rTable[src][dest] ← nb
21.         <congestion,congestedLinkSet> ← calculateLoad(network,traffic,rTable)
22.     else exit
23. until MAXITER iterations have been performed
```

Fig. 2. The Local Search RSNE algorithm

- Finally, the function *calculateLoad*(*network*,*traffic*,*rTable*) returns the network congestion given the network topology, the traffic pattern and the current routing scheme. The function also returns the set of links having maximum loads (*congestedLinkSet*).

The *candidateMoveSet* is empty at the beginning of each iteration. The local search algorithm (lines 3–23) consists of two parts. First, a set of alternative paths for some of the lightpaths passing through the most congested link is found (lines 6–17); in the second part (lines 18–22) a candidate is chosen and the corresponding routing table change is applied.

The first part (lines 6–17) includes the core of our proposal. The algorithm considers each congested link in *congestedLinkSet* (loop at lines 6-17). Then it iterates through all the routes using that link, identified by its endpoints (*cFrom*,*cTo*). Two nested loops are used: the first (line 7) scans the routing table of node *cFrom* looking for all destination nodes *dest* using that link; the second (line 8) scans all nodes *src* whose lightpaths directed to *dest* run through *cFrom*. These nodes identify the subtree rooted in *cFrom* of the routing tree having destination *dest*.

For each (*src*,*dest*) pair whose lightpaths go through the link (*cFrom*,*cTo*), the algorithm tries to reroute the lightpaths by altering the routing table in *src*. The corresponding load is temporarily removed from the current route (line 9), then an iteration through all neighbors *nb* of *src* calculates the maximum load that would be caused by rerouting the lightpath, provided that the new route does not end up in a cycle and that the congested edge is avoided. The best alternate paths, in terms of maximum load, are collected into *candidateMoveSet*. In particular, the current minimum is stored in *bestCandidateLoad*. If the load obtained after this traffic re-routing is equal to *bestCandidateLoad*, then the re-route is added to the candidate set (lines 12-13); if it is smaller, the candidate set is re-initialized to the current re-route and its load is stored as the new best value (lines 14-16). At the end of the alternate paths search, the partial load associated to the path originating in *src* and terminating in *dest* is reallocated (line 17) in order to allow the search of new paths with different initial nodes *src* (line 8).

In the second part of the RSNE algorithm, if the resulting set *candidateMoveSet* is not empty then one random element is selected from it (line 19), and the routing table of the network is updated (line 20). Finally, a new value of *congestion* and the corresponding set of most loaded links *congestedLinkSet* is calculated again in order to start a new search of alternate paths through the network.

Note that the local search algorithm continues looking for better values of *congestion* until the set of candidate re-routes *candidateMoveSet* is empty (line 22), or until a given number of iterations MAXITER has been performed (line 23).

From this algorithm we can easily obtain the RNE version: in this scheme, node *cFrom* is the only candidate for routing table modifications. This corresponds to the elimination of the loop structure on line 8, which scans the *cFrom*-rooted subtree, by setting *src* equal to *cFrom*. The rationale for RNE is to avoid a large tree exploration and to keep modifications as near as possible to the congested link. In fact, while rerouting at *cFrom* removes a whole bundle of lightpaths from the link, doing the same at some upstream node in the tree may cause a smaller reduction of the load. On the other hand, simulations in Sect. 4 show that, unless very few iterations are allowed before halting, performance of RNE is significantly worse than RSNE.

3 Incremental Implementation on Dynamically Evolving Traffic

Local search heuristics can be seen as stepwise refinements of an initial solution by slight modifications of the system configuration. In our case, the RSNE algorithm starts from a shortest path routing scheme and changes at every step a routing table entry of a single node in the matrix. By performing many such changes, the system reaches a minimal congestion configuration.

This iterative scheme suits in a very appropriate way to a dynamic environment where traffic requirements evolve with time. In particular, if changes in the traffic matrix are reasonably smooth[1] even a small number of steps of the RSNE algorithm in Fig. 2 is sufficient to keep the system in a suitable state as the traffic matrix changes. Of course, only lines 2-23 must be executed, because we don't want to restart from scratch by calculating the shortest path routing tables. Moreover, a very low number of iterations of the outer loop (lines 3-23) must be performed at each step, i.e. MAXITER must be small (1 to 5 should suffice) to avoid excessive traffic disruption. In the following, we

[1] The assumption is reasonable even though IP traffic is known to be bursty: in fact, traffic requirements are given as an average over a certain amount of time, with some marginal capacity left to accommodate traffic peaks.

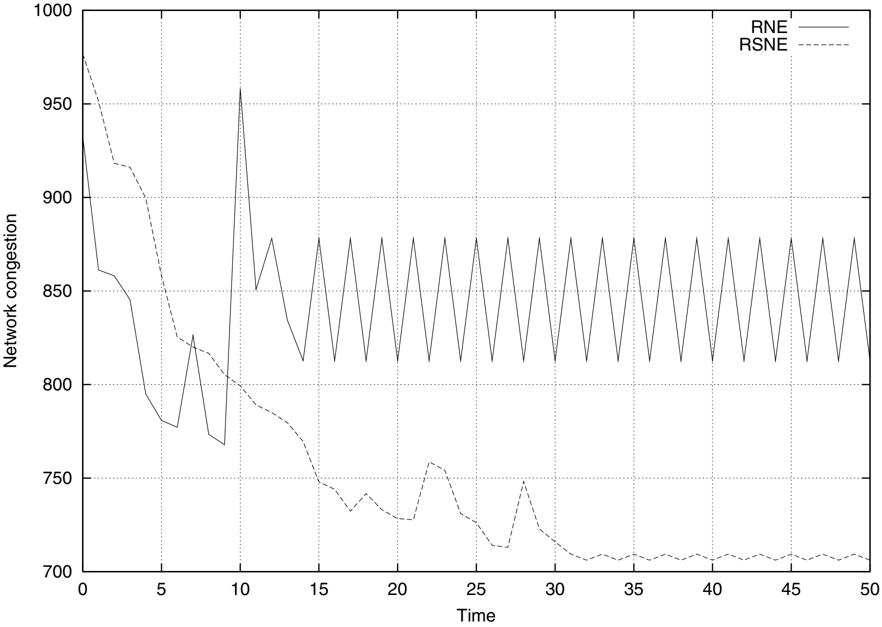


Fig. 3. Comparison between RSNE and RNE algorithms.

shall refer to the incremental algorithm as *Incremental* RSNE with k iterations per step: I-RSNE(k).

The simulations discussed in Sect. 4 show that even a single iteration of the algorithm yields good results under a fairly generic traffic model. The number of iterations of the algorithm is equal to the number of routing table entry modifications in the systems; thus, a very limited number of routing table entries must be modified as traffic evolves in order to keep congestion at low levels.

A similar approach has been proposed in [11], where branch-exchange methods are proposed for a local search heuristic; however, the type of local modification is quite different from our proposal.

4 Simulation Results

4.1 Static Traffic

To test the proposed algorithms we performed two sets of tests, static and dynamic. The first, using a static traffic matrix, explores the convergence speed of the RSNE and RNE algorithms.

Fig. 3 plots the evolution of the congestion value for a 50-iteration run of the RNE and RSNE algorithms with the same initial conditions; here the 14-node NSFNET backbone topology is used [10], while the traffic is randomly generated: every nondiagonal entry of the traffic matrix is a uniform value between 10 and 100. It turns out that the more complete RSNE algorithm outperforms its simplest version, although it sometimes achieves better results in the initial phase, probably because the algorithm is forced to move larger portions of load from edge to edge, achieving temporary better results but ending up with a complex, non-improvable routing scheme. Note that the congestion does not increase in a monotonic way: the algorithms do not halt when no improvement is possible, and the move leading to the smallest increase is chosen. This allows the system to escape local minima positioned in some shallow attraction basin. In many cases, this causes oscillation to take place once the minimum is achieved.

In the simulation shown here the maximum hop length corresponding to the lowest congestion configuration is 4. The corresponding value for the shortest path routing scheme is 3. The average hop length increases from 2.14 (shortest path) to 2.2 (RNE) and 2.21 (RSNE).

4.2 Dynamic Traffic

To investigate the behavior of the incremental version I-RSNE(k) with a dynamically evolving traffic pattern, we considered another topology, the 24-node regional network presented in [15].

To generate dynamic traffic we followed a model similar to that described in [9]. Given two positive integers N and Δ, we consider a sequence of $N\Delta+1$ traffic matrices $(T^0, T^1, \dots, T^{N\Delta})$ where matrix $T^{k\Delta}$, $k = 0, 1, \dots, N$ is random and independently generated. For each of these matrices a random maximum value between 10 and 100 is generated, and each entry of the matrix is calculated as a random number between 10 and this maximum. The random maximum value has been introduced to take into account the variability of internet traffic in the mid term. All other matrices are linear interpolations of the immediately adjacent random matrices. In other words, given $h = 0, \dots, \Delta - 1$ and $k = 0, \dots, N-1$, entry $T_{ij}^{k\Delta+h}$ of matrix $T^{k\Delta+h}$ is computed as follows:

$$T_{ij}^{k\Delta+h} = \text{round}\left[\left(1 - \frac{h}{\Delta}\right) T_{ij}^{k\Delta} + \frac{h}{\Delta} T_{ij}^{(k+1)\Delta}\right].$$

Fig. 4 describes the behavior of the proposed algorithms in the dynamic traffic case by comparing their congestion values. The upper plot represents the results achieved by the shortest path routing; for every traffic matrix, 50 different shortest path configurations were computed (with a random tie-breaking scheme), and the graph represents the $\mu \pm \sigma$ interval, where μ is the average and σ is the corresponding root mean square value. In fact, a large variability in the congestion (up to 35%) has been observed depending on random choices.

Note that all RSNE and I-RSNE results are almost equivalent, well under the shortest path values. The only difference can be seen in the initial transient, when the incremental versions begin to differ from the pure shortest path configuration. This is a very important feature of the algorithm, because I-RSNE(1) requires the modification of a single entry of the routing table of a single node for each change in the traffic conditions. The RSNE and I-RSNE algorithms achieve results that are 8% to 12% better than the shortest path minimum over all the 50 runs, and up to 32% better than the average shortest path result.

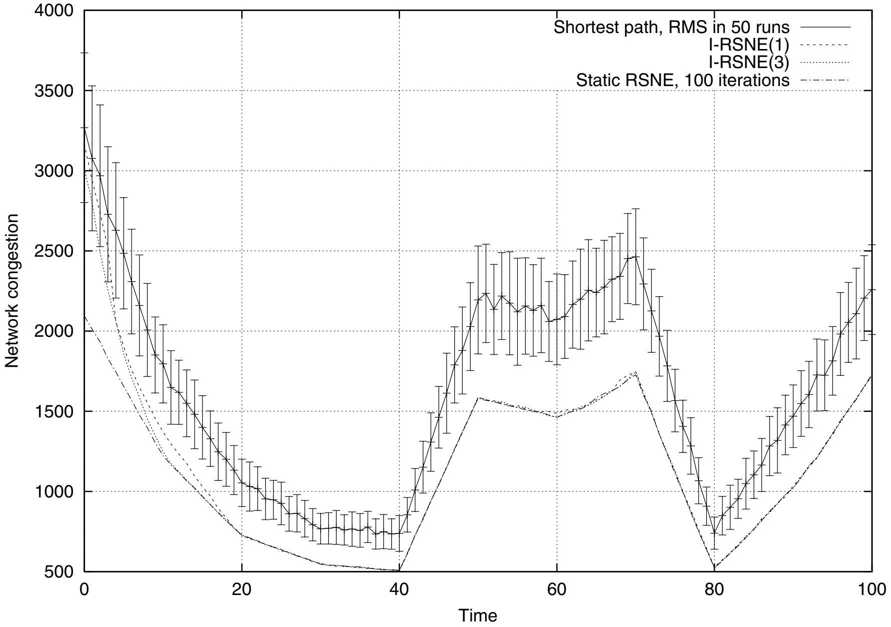


Fig. 4. Comparison in terms of congestion: shortest path, `RSNE` and `I-RSNE`(k).

If Fig. 4 is assumed to represent the traffic evolution during a day of real time, then a single change every fifteen minutes (in order to obtain about 100 changes per day) is sufficient to keep congestion at a local minimum, well below the shortest path routing.

Fig. 5 shows a comparison among the same algorithms in terms of average hop length, calculated as the mean value of hop distances (in the given routing scheme) between every node pair in the graph. The average hop length of shortest path routing, represented by the continuous bottom line, is obviously constant, and by definition it is the minimum (its value is 2.77).

The other plots, in particular the one representing the behavior of the offline `RSNE` algorithm, are particularly irregular when compared to those in Fig. 4; this is partly due to the narrower timescale, but it also depends on the fact that routing table changes are aimed at load reduction, and therefore hop lengths may vary from step to step. Note also that the `I-RSNE` outcomes are smoother, because adjacent results are strongly correlated, while the `RSNE` procedure performs a complete restart at every time step.

Fig. 5 highlights the main drawback of the incremental schemes `I-RSNE`(k): the shortest path configuration is never reimplemented, as was the case with `RSNE`, so the average hop length is slightly growing in time.

While `RSNE` is constantly above the shortest path value by about 2%, the `I-RSNE` schemes tend to accumulate longer paths, getting to a 7% increase after 1000 time steps. Note that the difference grows in time. To overcome the problem a simple modification consists of restarting from a shortest path configuration every time the average (or the maximum) hop length trespasses a given threshold.

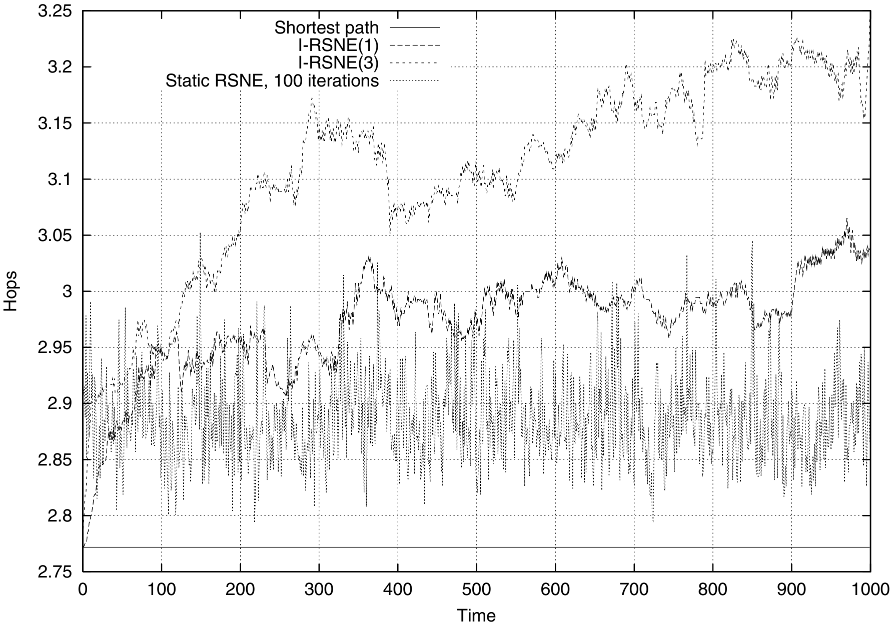


Fig. 5. Comparison in terms of average hop length: shortest path, RSNE and I-RSNE(k).

5 Conclusions

The paper proposed and motivated a heuristic technique for load balancing in IP-based optical networks (RSNE) built on simple modifications of routing tables. Some variations were introduced to reach lower algorithmic complexity (RNE) and to obtain a faster, incremental evaluation in the case of dynamically evolving traffic (I-RSNE).

Comparisons between the new techniques and the shortest path routing scheme, both in terms of network congestion and length of the resulting routes, show that the proposed algorithms are effective to reduce congestion, and outperform shortest path routing by up to 32%. The resulting increase in hop length is limited to a small amount (up to 7% in the worst case considered in the paper).

The RSNE algorithm explores all possible improvements before taking a step. Further investigation will determine how the quality of the solutions deteriorates if a randomized approach is followed in order to distribute the algorithm.

Acknowledgments. We would like to thank Imrich Chlamtac and Jason Jue of the University of Texas at Dallas, for their interesting and fruitful discussions with the authors about the subject of this work, and the anonymous referees for their precious comments.

References

1. I. Chlamtac, A. Ganz, and G. Karmi. Lightpath communications: A novel approach to high bandwidth optical WANs. *IEEE Transactions on Communications*, 40(7):1171–1182, 1992.
2. L. Fratta, M. Gerla, and L. Kleinrock. The flow deviation method: An approach to store-and-forward communication network design. *Networks*, 3:97–133, 1973.
3. M. Kodialam and T. V. Lakshman. Integrated dynamic IP and wavelength routing in IP over WDM networks. In *Proceedings of IEEE INFOCOM 2001*, pages 358–366, 2001.
4. J. Labourdette and A. Acampora. Logically rearrangeable multihop lightwave networks. *IEEE Transactions on Communications*, 39:1223–1230, August 1991.
5. Emilio Leonardi, Marco Mellia, and Marco Ajmone Marsan. Algortihms for the topology design in WDM all-optical networks. *Optical Networks Magazine*, 1(1):35–46, January 2000.
6. K. Lu, G. Xiao, and I. Chlamtac. Blocking analysis of dynamic lightpath establishment in wavelength-routed networks. In *Proceedings of ICC2002*, 2002. submitted.
7. D. Mitra, R. Gibbens, and B. Huang. State-dependent routing on symmetric loss networks with trunk reservations. *IEEE Transactions on Communications*, 41(2):400–411, 1993.
8. Ahmed Mokhtar and Murat Azizoğlu. Adaptive wavelength routing in all-optical networks. *IEEE/ACM Transactions on Networking*, 6(2):197–206, April 1998.
9. Aradhana Narula-Tam and Eytan Modiano. Dynamic load balancing in WDM packet networks with and without wavelength constraints. *IEEE Journal of Selected Areas in Communications*, 18(10):1972–1979, oct 2000.
10. R. Ramaswami and K. N. Sivarajan. Design of logical topologies for wavelength-routed optical networks. In *Proceedings of IEEE INFOCOM 1995*, 1995.
11. Jadranka Skorin-Kapov and Jean-François Labourdette. On minimum congestion routing in rearrangeable multihop lightwave networks. *Journal of Heuristics*, 1:129–145, 1995.
12. C. Xin, Y. Ye, T.S. Wang, and S. Dixit. On an IP-centric control plane. *IEEE Communications Magazine*, 39(9):88–93, 2001.
13. Bülent Yener and Terrance E. Boult. A study of upper and lower bounds for minimum congestion routing in lightwave networks. In *Proceedings of INFOCOM 1994*, pages 138–149, 1994.
14. H. Zang, J.P. Jue, L. Sahasrabuddhe, R. Ramamurthy, and B. Mukherjee. Dynamic lightpath establishment in wavelength routed networks. *IEEE Communications Magazine*, 39(9):100–108, 2001.
15. Zhensheng Zhang and Anthony S. Acampora. A heuristic wavelength assignment algorithm for multihop WDM networks with wavelength routing and wavelength re-use. *IEEE/ACM Transactions on Networking*, 3(3):281–288, June 1995.

Models for the Logical Topology Design Problem*

Nicolas Puech, Josué Kuri, and Maurice Gagnaire

École Nationale Supérieure des Télécommunications,
Computer Science and Networks Department,
46, Rue Barrault, 75632 Paris cedex 13, France
{npuech|kuri|gagnaire}@enst.fr

Abstract. We address the logical topology design problem (LTD) in WDM transport networks under static traffic assumptions. We start with one of the standard MILP formulations of the LTD problem that aims at optimizing the network congestion. We propose an improvement to this model that additionally optimizes the average hop count. We then derive a new MILP model that compels the traffic to be atomically routed. This last model enables fair comparisons of solutions obtained with MILP formulations and with metaheuristic algorithms. The latter allow us to deal with large size networks whereas the former are limited by their computational complexity. In this paper Tabu Search is used to tackle the LTD problem. We compare and discuss the logical topologies computed by the various methods described in the paper.

1 Introduction

The *network design problem* consists of defining an optimal network configuration that fulfills specified traffic demands under technical and economical constraints [1]. Optimality is typically defined as the minimization of the network cost, maximization of specific network performance parameters (e.g. availability, call blocking, average packet delay, network congestion, etc.) or maximization of a function of the performance/cost ratio. In general, the minimum set of inputs for the design and dimensioning problem consists of a forecast of the traffic demands, topological data describing the physical links between nodes, and technical and cost information of the equipment and transmission infrastructure. The resulting optimal network configuration is typically described in terms of information on

* **N. Puech is also with I.U.T., University Paris 5, 143 Avenue de Versailles, 75016 Paris, France, e-mail : puech@iut.univ-paris5.fr**. This work has been funded by Alcatel R & I, Marcoussis, France under Grant No. CI 578. Nicolas Puech has been funded by the French National Educational System (Éducation Nationale) and the University Paris 5 in the form of a sabbatical. Nicolas Puech would like to express his thanks to Mrs. Dominique Gascon (Director) and Mr. Serge Blumenthal (Statistics Department), both from the I.U.T., University Paris 5, France, for having decided in favor of his sabbatical. Josué Kuri has been partially supported by CONACyT Grant No. 122688.

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 301–313, 2002.

realization of traffic demands (e.g. working and protection paths of demands), location and configuration of equipment and network cost.

In *Wavelength Division Multiplexing* (WDM) transport networks, selective switching of wavelengths can be performed with current opto-electronic devices. Moreover, recent developments in optical devices make it possible to do this selective switching in the optical domain. From a network perspective, this functionality is of particular interest since it enables the development of networks that conveniently exploit the large transmission capacity of WDM systems. Clear optical channels, or *lightpaths*, that do not undergo opto-electronic conversion at intermediate nodes can be set between physically non-adjacent nodes in the network by assigning a wavelength to the lightpath and switching the wavelength optically. The information traveling on a lightpath is carried optically from end to end.

Because of equipment costs, fiber availability and switching capabilities at the network nodes, it is not possible to set up a full mesh of $N(N-1)$ lightpaths between the N nodes of a network. Thus, a particular subset of lightpaths, out of the set of all possible lightpaths, must be selected. This subset is called the *logical topology*, or virtual topology, seen by the electronic switches at the network nodes. The traffic demands are realized on top of the logical topology by routing them through direct lightpaths, when they exist between the source and destination of the traffic, or through a concatenation of lightpaths otherwise. The logical topology is realized by routing the subset of lightpaths over the physical topology[1] and assigning wavelengths to these lightpaths.

Network design problems are in general very complex because the resulting optimization problems are numerically intractable even for networks with a small number of nodes. This inherent complexity frequently leads to solution approaches based on the decomposition of global design problems into subproblems of tractable complexity; each identified subproblem can be solved by a separate algorithm. Thus, a solution to the global problem is found by sequentially solving the subproblems, i.e. the solution to a first subproblem (output) is the input to the next one. This approach has been adopted, for example, in the design of SONET/SDH networks [1] and has also been used in the framework of WDM network planning [2].

Following a decomposition approach, three subproblems of the network design problem are usually considered in the literature:

- LTD (Logical Topology Design): the definition of the lightpaths to be set as the virtual topology and the routing of traffic demands over the virtual topology,

- LR (Lightpath Routing): the routing of lightpaths of the virtual topology over the physical topology,

- WA (Wavelength Assignment): the assignment of wavelengths to lightpaths.

A drawback of the decomposition approach is that it precludes the enforcement of global optimality criteria because design choices leading to the optimality of one of the subproblems can be detrimental to the subsequent subproblems. It is possible to circumvent this drawback by first solving the subproblems sequen-

[1] or over a different server network layer.

tially and then executing a second resolution; this second time, design information of the first phase can be exchanged as *feedback* between the algorithms that solve the subproblems so that "bad" design choices are known a priori and, consequently, are avoided. We adopted this approach in the resolution of the problems addressed in this paper. We first solve the LR problem considering a full-meshed logical topology for the considered networks. We then use the information about the routed lightpaths to prune the variable space associated with the MILP description of the LTD problem. This allows us to suppress a priori a number of solutions already considered unsuitable for the LTD problem. The size reduction resulting from this pruning process allows us to tackle networks up to 15 nodes whereas the MILP description considering all possible solutions only works for networks up to 7 nodes[2]. The pruning process is fully described and analyzed in [4].

In the following section we define the LTD problem and its tradeoffs. We then describe in Section 3 a multicommodity flow model that provides a mathematical description of the LTD problem as a Mixed Integer Linear Program (MILP). The aim is to minimize the network congestion, that is, to minimize the value of the traffic passing on the most congested lightpath. We show how the model may be improved in order to additionally minimize the average hop count. This model allows the traffic to be arbitrarily split over the chosen lightpaths. In Section 4 we derive a new model that enforces atomically routed traffic, that is, the traffic demand of one source-destination pair must follow the same route as a whole. The considered MILP models are intractable even for networks of small size which makes it often impossible to compute an exact solution to the LTD problem. When considering real size networks, one may be led to accept only approximate solutions to the problem, provided these are not too far from the exact solutions. Metaheuristic methods compute such approximate solutions for large size networks. As such, they are an interesting alternative to MILP based models. In Section 5, we describe a variant of the Tabu Search (TS) metaheuristic that we used for our study. TS is commonly used in optimization, particularly in the field of network planning [5,6]. Like the model described in Section 4, TS enforces atomic routing of the traffic.

In Section 6 the solutions computed by these various algorithms are compared. We first evaluate the improvement brought by the hop count minimization. In order to evaluate the quality of the solutions computed by TS, it is interesting to compare them to the ones computed by the two preceding MILP models. The first model provides a lower bound on congestion, whether the routing is atomic or not. The second model provides a fair comparison since it takes into account atomic routing as is the case with TS. We will then compare the results computed by the three algorithms described previously. We will notice that TS reaches often the best possible solution. We hence legitimize the use of TS for large size networks.

[2] We used the OSL ILP solver [3] on a Sun Ultra 5 workstation with 128 Mbit RAM for our simulations. We limited our experiments to 10^8 iterations of the solver. In most cases an optimal solution was found before reaching this limit.

2 Logical Topology Design

The design of a logical topology and the routing of traffic demands over the topology can be formulated as an optimization problem. Optimality criteria are typically defined in terms of either network implementation cost or performance metrics [7]. Moreover, the constraints of the problem usually come from hardware limitations imposed by transmission and switching equipments.

Logical topology design in WDM transport networks has been extensively investigated in recent years [2,5,8,9,10,11]. Most of the proposed models aim at optimizing network performance metrics such as network congestion, throughput and average transit delay, rather than cost metrics. Focus on network performance criteria stems from the fact that the logical topology is the network layer directly seen by the electronic switches, which are typically expensive and, most importantly, offer *limited* capacity when compared to optical switches.

The logical topology design involves two tradeoffs: one between performance and network implementation costs and another one between capacity allocation in the electronic switches and the optical switches. Thus, a logical topology must be designed to meet performance and network cost criteria while satisfying hardware constraints at both the electronic and optical switching levels.

3 Independently-Routed Multicommodity Flow Model

Some multicommodity flow models have been developed to solve the LTD problems by mathematical means. Some of these models aim at solving the LTD, LR and WA problems simultaneously [11], while others consider either, LTD and LR [10] together, or LTD alone [8,9]. Integrated models that jointly consider the three mentioned problems are expected to yield better solutions[3] than models that address the problems separately. However, the complexity of integrated models precludes their use when solving problem instances of realistic size. The two multicommodity flow models presented in this paper consider the LTD problem alone. The LR and WA problems are not considered here though, as mentioned previously, we will prune the LTD variable space according to the feedback method described in [4].

We begin the presentation of the first multicommodity flow model by introducing the notation. We will keep the same notation for the second model in Section 4 and for the application of tabu search in Section 5.

Indices:

i,j used as subscripts, denote the source and destination nodes of a lightpath.

s,d used as superscripts, denote the source and destination nodes of a traffic demand.

[3] with respect to specific global network performance and resource utilization criteria.

Problem parameters:

N the number of nodes in the network (the nodes are numbered $1, 2, \ldots, N$).

(λ^{sd}) an $(\mathbb{R}_+)^{N \times N}$ matrix describing the amount of traffic flowing from each source node s to each destination node d, expressed in some convenient units (e.g. Mbit/s). The matrix is not necessarily symmetrical.

δ_{out} the logical out-degree of the nodes in the network, i.e. the maximum number of lightpaths that can initiate at the electronic switch of the network nodes.

δ_{in} the logical in-degree of the nodes in the network, i.e. the maximum number of lightpaths that can be terminated at the electronic switch of the network nodes. It is often assumed that $\delta_{out} = \delta_{in} = \delta$, although this is not a strict requirement.

Variables:

b_{ij} a binary variable that indicates whether or not there is a lightpath from source node i to destination node j (1 and 0 respectively). Note that lightpaths are directed, therefore, $b_{ij} = 1$ does not imply $b_{ji} = 1$.

λ_{ij}^{sd} a real variable indicating the amount of traffic from source s to destination d passing through the lightpath going from i to j.

Computed values:

λ_{ij} the total amount of traffic passing through the lightpath going from node i to node j, i.e. $\lambda_{ij} = \sum_s \sum_d \lambda_{ij}^{sd} \; \forall\, i, j$.

λ_{max} the amount of traffic passing through the most congested lightpath, i.e. $\lambda_{max} = \max_{1 \le i,j \le N} \lambda_{ij}$. This term is referred to as the *maximum congestion level* in the network or simply, the *congestion.*

Model MILP1

This multicommodity flow model is partially based on the formulation presented in [8]. The model is defined in terms of b_{ij} (binary) and λ_{ij}^{sd} (real) variables, hence it is a MILP (Mixed Integer Linear Program). Once the problem is solved and the values of the variables are known, the variables b_{ij} set to 1 indicate the lightpaths to be established and, for each traffic demand λ^{sd}, the values of the variables λ_{ij}^{sd} indicate the amount of *sd* traffic flowing on lightpath ij. The model is:

$$\textbf{Minimize: } \lambda_{max} \tag{1}$$

Subject to:

Flow conservation at each node:

$$\sum_i \lambda_{ij}^{sd} - \sum_k \lambda_{jk}^{sd} = \begin{cases} -\lambda^{sd}, \text{ if } j = s \\ \lambda^{sd}, \quad \text{if } j = d \\ 0, \qquad \text{otherwise} \end{cases} \quad \forall\, j, s, d \tag{2}$$

Total flow on a lightpath:

$$\lambda_{ij} = \sum_s \sum_d \lambda_{ij}^{sd} \leq \lambda_{max}, \forall\, i, j \tag{3}$$

$$\lambda_{ij}^{sd} \leq b_{ij} \lambda^{sd}, \qquad \forall\, i, j, s, d \tag{4}$$

Degree constraints:

$$\sum_j b_{ij} \leq \delta_{out}, \forall\, i \tag{5}$$

$$\sum_i b_{ij} \leq \delta_{in}, \forall\, j \tag{6}$$

Value range constraints:

$$\lambda_{ij}^{sd} \geq 0, \quad \forall\, i, j, s, d \tag{7}$$

$$b_{ij} \in \{0, 1\}, \forall\, i, j \tag{8}$$

The objective function (1) states that the model aims at minimizing λ_{max}, the *maximum congestion level* in the network. Equation (2) is the flow conservation constraint which states that, for a traffic demand λ^{sd}, at every intermediate node j ($j \neq s, j \neq d$), the amount of flow entering the node must be equal to the amount leaving it. At the source node ($j = s$), the traffic only leaves the node, whereas at the destination node ($j = d$), the traffic only enters the node. Equation (3) ensures that the total amount of traffic passing through any lightpath is at most equal to λ_{max}. Equation (4) states that traffic can flow on a lightpath only if the lightpath exists. Moreover, for a traffic demand from s to d, the amount of traffic flowing on the lightpath from i to j cannot be more than the total amount of the traffic demand. Equations (5) and (6) enforce the maximum logical out-degree and in-degree of the nodes, respectively. Note that some nodes with degree *smaller* that δ can be accepted in the solution. As we shall see later, this allows us to find virtual topologies that minimize the congestion while using as few lightpaths as possible. Finally, equations (7) and (8) constrain the flow indicator variables to be non-negative reals and the lightpath indicator variables b_{ij} to be binary.

Model MILP1b

It may happen that multiple solutions minimize the congestion for the same problem instance. Thus, once the minimum possible congestion value λ_{max}^* has been found, one can look within the set of solutions for a solution that optimizes a different criterion. For example, a solution that minimizes the average hop count (the average number of lightpaths that the traffic demands traverse to go from source to destination) for a given λ_{max}^* can be found by modifying the preceding formulation in the following manner:

- first, replace the objective function (1) by

$$\textbf{Minimize: } \sum_i \sum_j \sum_s \sum_d \lambda_{ij}^{sd} \tag{9}$$

- then, replace the variable λ_{max} in equation (3) by the value λ^*_{max}.

This leads to a two-step optimization process. The improvement of this model with respect to the basic one is assessed in Section 6, where a comparison between the results obtained in step 1 and the ones computed in step 2 is provided.

4 Atomically-Routed Multicommodity Flow Model

The previous model allows a traffic demand λ^{sd} to be arbitrarily split across multiple possible routes. While this property is useful to determine the lower bound of the *congestion* in the network, the *routing* of traffic demands obtained in this manner may have little practical relevance. Though existing packet-switched networks (e.g. ATM, Internet) provide a fine-grained switching granularity, this granularity cannot be used to arbitrarily split a traffic demand because of restrictions imposed by network protocols (e.g. avoiding packet reordering in TCP connections) or architectures.

We propose a model that allow the traffic demands to be *atomically routed*, i.e. the whole traffic from source s to destination d follows the same route through the network, as suggested in [10]. Metaheuristic methods usually assume atomic traffic routing, and it is fair to compare their solutions with the ones computed with this model rather than with the ones that allow traffic splitting like the model MILP1.

Model MILP2

The model is directly inferred from the model MILP1 with the introduction of new variables $\varphi_{ij}^{sd} = \lambda_{ij}^{sd}/\lambda^{sd}$ to describe the flow and an additional constraint to avoid traffic splitting: $\varphi_{ij}^{sd} \in \{0,1\}, \forall\, i,j,s,d$. Thus, the model becomes:

$$\textbf{Minimize: } \lambda_{max} \tag{10}$$

Subject to:
constraints (2), (3), (4), (5), (6) and (8).

Model MILP2b

As in model MILP1, once the congestion value λ^*_{max} has been found, the objective function (10) can be replaced by

$$\textbf{Minimize: } \sum_i \sum_j \sum_s \sum_d \varphi_{ij}^{sd} \tag{11}$$

in order to find a routing of demands that additionally minimizes the average hop count.

5 Tabu Search: A Metaheuristic Approach

The major drawback of multicommodity flow models is that they are computationally intractable even for small problem instances. Optimization algorithms for the LTD problem based on metaheuristics have recently gained interest because they are able to cope with problem instances of large size. Among others, simulated annealing [6], and Tabu Search (TS) [5] have been proposed to tackle the LTD problem. The computed solution however is not guaranteed to be optimal. These algorithms have the same aim as the multicommodity flow model presented in Section 4 (model MILP2) in the sense that they try to find a virtual topology with minimal congestion under given logical degree constraints and using atomic routing of traffic demands.

Tabu search has proven to be an effective metaheuristic method for LTD [5,6]. This section describes an optimization algorithm for the LTD problem based on TS that was initially proposed in [5]. A detailed description of the TS algorithm can be found in [12].

We will summarize TS as follows. TS consists of exploring the space of solutions until a number of iterations is reached or until a specific cost criterion is satisfied. The exploration starts with an initial solution computed by another algorithm (among others, initial solution randomly choosen). At each iteration, TS computes a set (called neighborhood) of solutions derived from the current solution via perturbations applied to this solution. All the solutions of the neighborhood are evaluated (see below) and the best one is selected as the new current solution. In order to prevent the algorithm from cycling along the same series of current solutions, a tabu list is maintained. It contains a number of last visited solutions, which cannot be chosen as long as they belong to this list. This allows the algorithm to choose a solution worse than the current one, allowing it to escape from the local minima encountered during the search.

6 Experimental Results

An instance of the LTD problem is defined by a *physical topology* and a *traffic matrix* (λ^{sd}). The physical topology corresponds to the nodes of the network and the fiber links connecting them.

The elements of the traffic matrix represent a forecast of the packet traffic demand between any two nodes in the network expressed in generic units. The solution to the problem is a set of lightpaths that constitute the virtual topology.

An *optimal* solution is one that minimizes the traffic congestion in the virtual topology. For the experiments of this Section, we use a physical topology that corresponds to a hypothetical backbone network of 9 nodes.

Concerning the traffic demands, in a first step, we used two different traffic matrices. The first one, hereafter referred to as Matrix1, comes from a real traffic evaluation and has a particular element element which is much larger than the others and whose value is 847 traffic units. This value is the lowest bound that any atomic routing algorithm may reach for congestion. The average value of Matrix1 is 123.63 traffic units and the smallest value is 1. In the second traffic matrix, that we call Matrix2, all the elements, except those of the diagonal, are $\lambda^{sd} = 124.0 \quad \forall s, d \quad s \neq d$. The purpose of this second matrix is to evaluate the results of the algorithms under the particular condition arising when all the nodes in the network exchange the same amount of traffic. Thus, we have two different problem instances: one represented by the physical topology and Matrix1, and one represented by the same topology and Matrix2.

Table 1. Congestion (λ_{max}) obtained with MILP1, MILP2 and TS.

	Matrix 1			Matrix 2		
Degree	MILP1	MILP2	TS	MILP1	MILP2	TS
3	752.16	(923.00)	847.00	620.00	620.00	620.00
4	524.00	847.00	847.00	372.00	496.00	496.00
5	492.00	847.00	847.00	286.15	372.00	496.00
6	492.00	847.00	847.00	233.58	248.00	372.00
7	492.00	847.00	847.00	201.50	248.00	327.00

Table 1 presents[4] the congestion of the logical topologies obtained with MILP1, MILP2 and TS, respectively for Matrix1 and Matrix2 traffic demands, for degree values ranging from 3 to 7. It is noticeable that TS always reaches the best possible value for Matrix1. Let us recall that TS, like MILP2, enforces atomic routing, that is, the whole traffic demand of a source-destination pair must follow the same route. Hence, in the case of atomic routing, the optimal congestion value is the greatest value of the traffic matrix. When the degree is 3, TS even outperforms MILP2. Indeed, the value indicated in that case for MILP2 is the partial result obtained with the solver since the solver did not manage to find the optimal solution 847 before the maximum allowed number of iterations.

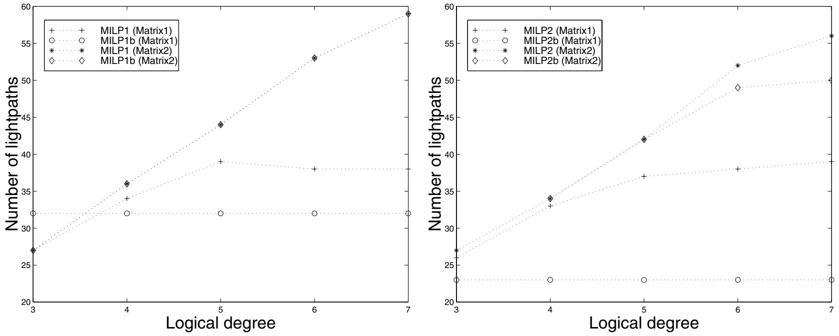
In the case of Matrix2, the difference in the congestion values computed by MILP1 and MILP2 is not so important as in the case of Matrix1. TS is quite good for degree 3 or 4 but reaches values higher than the ones found by MILP2. It seems that the case of a homogeneous traffic demand is more difficult to tackle.

It is also interesting to observe that allowing traffic to be independently routed allows us to reach far better congestion values in any case. This suggests that it might be interesting to study traffic splitting issues in future research.

Finally, in the case of Matrix1, it appears unnecessary to study logical topologies with degree higher than 3 since the lower bound for congestion is already found by the atomic routing algorithms, MILP2 or TS, when the degree is 3. Conversely, in the Matrix2 case, the congestion is significantly reduced when the

[4] The value indicated for Matrix1, MILP2 and degree 3 was the one computed at the iteration limit (see footnote 3) and is not optimal.

degree increases. However, increasing the logical degree increases the number of necessary lightpaths and, ultimately, the number of ports in the optical switches.



(a) MILP1 and MILP1b

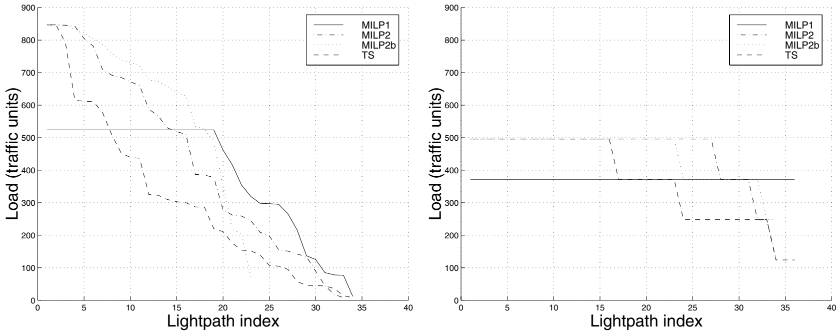
(b) MILP2 and MILP2b

Fig. 1. Required number of lightpaths when using MILP1, MILP1b, MILP2 and MILP2b models as a function of the logical degree (with Matrix 1 and with Matrix 2).

The results collected with Matrix1 seem to indicate that it has a very special traffic distribution hence leading to extreme results (among others, the maximal value of Matrix1 being far higher than the mean value, both algorithms reach this bound). That is why we had another series of experiments to compare the algorithms. We generated 10 traffic matrices randomly with uniformly distributed values in the range $0 - 1000$. We computed the congestion obtained for these matrices with MILP1, MILP2 and TS and observed the following results. Note that almost all computations lead with MILP2 had to be stopped due to iteration limit, hence the values obtained were not optimal. The average congestion (respectively standard deviation) obtained is 1726.2 (141.8) for MILP1, 2590.5 (396.0) for MILP2 and 1997.9 (152.9) for TS. The mean congestion increase from MILP1 to MILP2 is 50% and it is 16% from MILP1 to TS. This last result clearly shows the practical limit of MILP2 since TS allows us to get better values than the interrupted MILP2 search in much less time.

Figure 1 shows the required number of lightpaths computed by MILP1/ MILP1b and MILP2/MILP2b models as a function of the logical degree for both Matrix1 and Matrix2. The number of required lightpaths is smaller when using the two-step optimization MILP1b/MILP2b variants of the models. This result was expected since both MILP1b and MILP2b minimize the congestion and the average hop count. The difference between MILP1/MILP2 and their variants MILP1b/MILP2b is more important in the case of Matrix1, when the traffic demand is non homogeneous. Conversely, one can notice that in the case of a homogeneous traffic demand (Matrix2), the improvement of MILP1b/MILP2b is not very significant (it is null in the case of MILP1b) whereas there was a gain of up to 40% in the number of lightpaths in the case of Matrix1 with MILP2b.

Figure 2 shows the traffic distribution over the lightpaths computed by MILP1, MILP2, MILP2b and TS, for a logical topology with degree 4 and for Matrix1 and Matrix2. It is clear that MILP1 outperforms the other algorithms since it is allowed to split the traffic over several routes. The comparison can only be considered as fair between MILP2, MILP2b and TS, and clearly the curves corresponding to these methods are close and of the same shape. One can also notice on Figure 2 that MILP2b not only requires a smaller number of lightpaths to establish the traffic demand, but also computes a better lightpath usage : the 20 first lightpaths support more traffic than the corresponding ones computed by MILP2 or TS. MILP2b can be considered as more efficient in this respect.



(a) Matrix 1

(b) Matrix 2

Fig. 2. Distribution of traffic load on the lightpaths of the virtual topology (degree=4) computed by MILP1 (which stands as a lower bound for congestion), MILP2, MILP2b and TS. Tests for Matrix1 and Matrix 2.

This series of experiments presents a fair comparison between a MILP model and TS and shows that the results obtained with TS are close to the ones computed with the other models. While one should not underestimate the difference that remains between TS and MILP2 (a 30% congestion increase in the case of Matrix2, which is a kind of worst case), it is worth mentioning that in our experiments TS was 100 to 10000 times faster than the tested MILP models. Moreover, TS is able to compute solutions to the LTD problem for networks with as much as 150 nodes whereas MILP models become intractable for more than 15 nodes. The results obtained in this study plead in favor of TS which provides results fairly close to the optimal solution with far better computing times on the tested situations. It hence seems reasonable to compute approximate solutions to the LTD problem with TS for real size networks. Moreover, it might be interesting to modify the TS algorithm so that it can additionally perform the average hop count minimization, since MILP1b and MILP2b show an improvement with respect to MILP1 and MILP2.

7 Conclusion

Starting from a standard multicommodity flow model describing the LTD problem, we have proposed a second optimization step in order to find solutions that additionally minimize the average hop count. Experiments have shown that this two-step approach leads to solutions requiring fewer lightpaths than the solutions found by the original model. We also derived a new model that enforces atomic traffic routing. This models serves as a base to compare solutions to the LTD problem obtained with MILP formulations and a TS metaheuristic. Experiments have shown that TS often reaches one of the best possible solutions. Hence TS, even though it is a method which does not ensure that one will find an optimal solution, may be a very good alternative to solve the LTD problem, especially since it enables one to deal with much larger problem instances than MILP based models which are computationally intractable. An improvement to the current TS LTD algorithm would consist of modifying the cost function in order to additionally minimize the average hop count, as done in the MILP models presented in the paper.

References

[1] Eurescom. Planning of Optical Networks. Deliverable 3 Project P709, Eurescom, 2000.

[2] E. Leonardi, M. Mellia, and M. Ajmone Marsan. Algorithms for the logical topology design in WDM all-optical networks. *Optical Networks Magazine*, 1(1):35–46, 2000.

[3] IBM Spatial and Optimization Solutions. *IBM's Optimization Solutions and Library, version 3.*

[4] J. Kuri, N. Puech, and M. Gagnaire. Resolution of a WDM optical network design problem using a decomposition approach and a size reduction method. ECUMN 2002, Colmar, France, 2002.

[5] M. Mellia, A. Nucci, A. Grosso, E. Leonardi, and M. Ajmone Marsan. Optimal design of logical topologies in wavelength-routed optical networks with multicast traffic. In *Proceedings of INFOCOM 2001*. IEEE Communications Society, 2001.

[6] R. Caberlon, W. Floris, and N. Puech. Logical topology design : a case study. Private communication, École Nationale Supérieure des télécommunications, 2001.

[7] M. Ammar, S. Cheung, C. Scoglio, I. Chlamtac, A. Faragó, and T. Zhang. Virtual path network design. In J. Roberts, U. Mocci, and J. Virtamo, editors, *Broadband Network Teletraffic. Final report of action COST 242*, pages 271–299. Springer-Verlag, Berlin Heidelberg, 1996.

[8] R. Ramaswami and K. N. Sivarajan. Design of logical topologies for wavelength-routed optical networks. *IEEE Journal on Selected Areas in Communications*, 14(5):840–851, 1996.

[9] B. Mukherjee, D. Banerjee, S. Ramamurthy, and A. Mukherjee. Some principles for designing a wide-area WDM optical network. *IEEE/ACM Transactions on Networking*, 4(5):684–696, 1996.

[10] D. Banerjee and B. Mukherjee. Wavelength-routed optical networks: Linear formulation, resource budgeting tradeoffs, and a reconfiguration study. *IEEE/ACM Transactions on Networking*, 8(5):598–607, 2000.

[11] R. Krishnaswamy and K. Sivarajan. Design of logical topologies: A linear formulation for wavelength-routed optical networks with no wavelength changers. *IEEE/ACM Transactions on Networking*, 9(2):186–198, 2001.

[12] F. Glover and M. Laguna. *Tabu Search.* Kluwer Academic Publishers, Boston, MA, 1997.

Dynamic Shaping for Self-Similar Traffic Using Network Calculus

Halima Elbiaze, Tijani Chahed, Tülin Atmaca, and Gérard Hébuterne

Institut National des Télécommunications
9 rue Charles Fourier 91011 Evry CEDEX - France
{halima.elbiaze, tijani.chahed, tulin.atmaca,
gerard.hebuterne}@int-evry.fr
Phone : +33 1 60 76 47 42 , Fax : +33 1 60 76 47 80

Abstract. The focus of this paper is the shaping of self-similar traffic at the access of an optical node. We propose a novel algorithm that dynamically shapes the incoming traffic, based on service curves equations, in order to meet the optical nodes constraints in terms of buffer size or delay. We first estimate arrival parameters within various time intervals in order to make the incoming traffic fit into a token bucket traffic specification (Tspec) format. We then derive the shaping parameters based on deterministic service curves. Those shaping parameters vary dynamically according to the Tspec of every time window. We eventually set those parameters back into the original model in order to meet some QoS constraints at the optical network level.

1 Preliminaries and Problem Relevance

Optics has been identified as a key technology able to provide a large capacity to transport massive IP flows, and to cope with different Quality of Service (QoS) requirement. The self-similar nature of IP traffic has been demonstrated by several studies and mesurements. Due to the lack of optical memories, QoS could be offered through combined exploitation of electronic memories in the edges and optical ressources in the core of the optical network. The traffic shaping takes place in the edges and has a real impact to maintain the logical performance to its highest level.

Both IETF, and ITU have identified traffic shaping as a way to: 1) allocate a suitable amount of resources (buffer memory, bandwidth) to a connection to achieve its required QoS and 2) police traffic and assure ¨fair¨ access to a shared resource. The problem studied in this paper [1], is motivated by the desire to obtain applicable performance bounds for a very high-speed optical network dealing with self-similar traffic. One may view the problem considered in this paper as a ¨channel capacity¨ issue associated with dynamic shaping at the network edge.

[1] supported by the "Réseau National de Recherche en Télécommunications" under the decision number 99S 0201-0204 and the European IST Project DAVID (Data and Voice Integration over DWDM)

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 314–326, 2002.

To achieve this aim, a tool for studying end-to-end, bounded performance is needed. Classical queuing analysis studies average performance for aggregate traffic. It focuses on single server environments with attractive traffic models. However, in the packet-switching, integrated services models, bounds on the end-to-end performance need to be studied in a networking environment with traffic dynamics, interactions and burstiness far more complex than in the previous case. Worst-case performance bounds on the packet flow make it possible to derive guaranteed maximum and minimum values rather than averages; which is necessary when dealing with emerging multimedia networking scenarios. In this work, we use the deterministic version of the service curves method [3] and particularly Network Calculus (NC) [1], its Min-Plus algebra formulation. The remainder of this paper is organized as follows. In Section 2, we set the end-to-end system and describe the modeling of source and network in terms of arrival and service curve. In Section 3, we study the performance of the system in the absence of shaping. We also motivate the need for dynamically shaping the traffic in order to meet the network constraints. In section 4, we focus on dynamic shaping between a LAN and the optical network, we show the regions of shaping to meet both buffer and delay constraints and propose a novel algorithm for dynamic shaping based on the equations of the service curves. Some numerical results are presented and discussed in Section 5. Concluding remarks are eventually given in Section 6.

2 End-to-End System

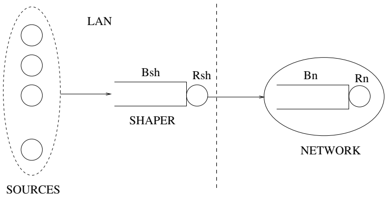


Fig. 1. End-to-end System

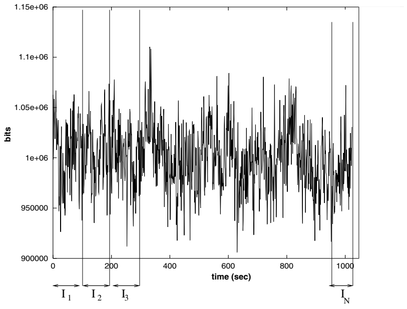


Fig. 2. Arrival process

2.1 Source Modeling and Arrival Curves

In source modeling, packet and connection arrival processes are often assumed to be Poisson owing to the attractive theoretical properties of such models [6]. Numerous studies have shown, however, that for both LAN and WAN networks, the distribution of packet interarrivals clearly differs from the exponential distribution [5]. Recent works argue convincingly that LAN traffic is much better modeled using self similar processes [11], which have very different theoretical properties than Poisson models. A subsequent investigation suggests that the same holds for WAN traffic too [12].

The strength of self similar models is that they are able to incorporate Long-Range Dependence (LRD), which informally means significant correlations across arbitrarily large time scales. For many networking issues, the presence or absence of LRD plays a critical role in the behavior predicted by analytical models. For example, the presence of LRD can completely alter the waiting times at the tail of a queue [4]. Self-similar processes are very difficult to tackle and render traffic characterization cumbersome in this case. One way to circumvent this is to bound the traffic rather than exactly characterizing it as suggested by recent models based on the service curves approch [7], [2], [8], [9] and [10]. Explicitly, in [7], a traffic stream is said to satisfy the $(X_{min}, X_{ave}, I, S_{max})$ model if the inter-arrival time between any two packets in the stream is larger than X_{min} during any interval I of length l, the average packet inter-arrival in I is larger than X_{ave}, and the maximum packet size is smaller than S_{max}. Alternatively, referring to [2], a traffic stream satisfies the (σ, ρ) model if, in I, the number of bits is less than $\sigma + \rho u$, $u \in I$. In the (σ, ρ) model, σ and ρ can be viewed as the maximum burst size and the long term bounding rate of the source, respectively. A similar argument is used in [8] and [9]. Rather than using the bounding rate, the Deterministic Bounding Interval-Dependent (D-BIND) model, found in [10], uses a family of rate-interval pairs where the rate is a bounding rate over the corresponding interval length. The model captures the intuitive property that over longer interval lengths, a source may be bounded by a rate lower than its peak rate and closer to its long-term average rate.

Traffic Specification (Tspec), introduced by the IETF for IP, is a description of the allowed traffic pattern a source can emit and not the actual one. A pair of token buckets is used by the traffic sender to describe the traffic it expects to generate and by the QoS control services to describe the parameters of traffic for which the reservation should apply. A token bucket specification is not a characterization parameter but a data structure definition. It takes the form of a token bucket, with rate r and depth b, plus a peak rate p and maximum packet size M. Units are bytes and bytes per second. We now show how self-similar traffic can be made to fit into a token bucket Tspec format.

Figure 2 depicts a self-similar process. As stated earlier, it is characterized by a similar irregular behavior at different time scales. To put such a process in a Tspec formulation, we partition the whole process into N equal, non-overlapping blocks, corresponding to time intervals $(I_i)_{i=1,\ldots,N}$, of length l_N, and approximate the traffic volume within each interval I_i by a corresponding set of Tspec parameters. In doing so, we obtain a piece-wise Tspec formulation of the global process which approximates the actual process. Let us note that the piece-wise decomposition of the entire process shall yet reflect its self-similar nature. The obtained Tspec in each interval seems to be indeed correlated, as will be illustrated in the simulations. Building on the arguments found in [10], and depending on the value of l_N, one can bound the traffic volume within I_i by either of three

ways. One, for small l_N, i.e., N large, a peak rate bound is sufficient. As l_N gets larger, i.e., N smaller, a peak rate only is too conservative and one needs to refine the volume bound by incorporating a mean bound too which gives the second formulation in terms of peak and mean rates. Three, for l_N on the order of the duration of the global process itself, a mean rate bound may be sufficient.
Using the service curve approach, an arrival function x is said to be bounded by an arrival curve α for all t if and only if for all $s < t$, $x(t) - x(s) < \alpha(t - s)$. Graphically, Figure 3 depicts the arrival curve corresponding to the Tspec given in terms of p, M, r, b. $\alpha(t)$ equals in this case $\min(pt + M, rt + b)$. Thus, in each interval I_i of length l_N, α_i^N is an arrival curve which takes one of those forms:

$$\alpha_i^N(t) = \begin{cases} r_i^N t & \text{for } t \in I_i \ , l_N = L/N \ \text{ and } N \text{ small} \\ min(p_i^N t, r_i^N t) & \text{for } t \in I_i \ , l_N = L/N \ \text{ and } N \text{ medium} \\ p_i^N t & \text{for } t \in I_i \ , l_N = L/N \ \text{ and } N \text{ large} \end{cases}$$

The very values of ¨large¨, ¨medium¨ and ¨small¨ depend on the arrival process itself and the desired accuracy of our bounding approximation. For a highly self-similar process, given by a high value for the Hurst parameter, the intervals I_i should have small lengths l_N. In this case, the peak rate bound applies better as an approximation. If on the contrary, the process is not highly correlated, given by a small Hurst parameter, a mean rate bound offers a good approximation. For the other cases, a peak plus mean formulation applies.

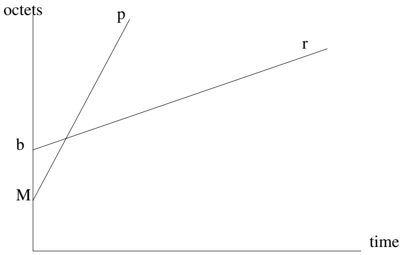


Fig. 3. Arrival curve - Tspec

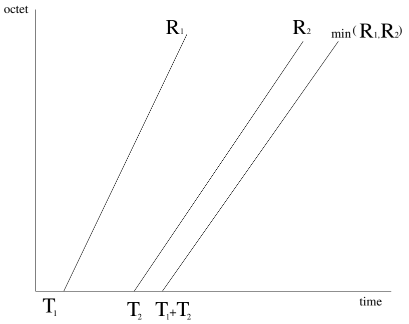


Fig. 4. Concatenation

2.2 Network Modeling and Service Curves

Using the service curve approach, a service curve is defined as the minimal service offered by a single server to our arrival curve. Analytically, for an input function x and output function y, β is a service curve if and only if for all $t \geq 0$, there exists some $t_0 \leq t$ such that $y(t) - x(t_0) \leq \beta(t - t_0)$. The IETF service curve has the form $\beta(t) = R(t - T_0)$ where R is the service rate and T_0 is the time at which the server starts serving our arriving traffic. A nonzero T_0 reflects the presence of a background traffic being served prior to our arriving traffic.

A network is not more than a cascade of successive network elements or servers each offering a service curve β_i. To ease the modeling and analysis of the network, the latter, i.e. the successive nodes that the traffic shall traverse, may be replaced by a single server reproducing their individual services as a concatenation of the individual servers. For n network elements in tandem, each one with service rate R_i and starting service at time T_i, $i = 1, .., n$, one possible concatenation scenario is a network element with a service curve $c(t) = R_n(t - T_n)^+$ where $R_n = \min(R_1, R_2, .., R_n)$ and $T_n = \sum_i^n T_i$, as shown in Figure 4.

3 Performance without Shaping

Let us recall that, for each interval I_i of length l_N, the input function is bounded by an arrival curve of the form $\alpha_i(t) = \min(p_i t + M_i, r_i t + b_i)$. The service curve at the network level is given as $c(t) = R_n(t - T_n)^+$, with service rate R_n and starting service at T_n, as the minimal service curve guaranteed by the server. Taking $c(t)$ as our actual service curve, the output curve is: $\alpha^*(t) = \min(R_n(t - T_n), rt + b)$ for $t > T_n$, as shown in Figure 5. According to the fundamental bounds of the service curve theory (maximum delay d_{max} and backlog B_{max}) in general, and Network Calculus [1] in particular, for $\theta = \frac{b-M}{p-r}$,

$$d_{max} = \frac{p - R_n}{R_n}\theta + \frac{M}{R_n} + T_n \text{ at } t = \theta \text{ if } p > R_n > r \tag{1}$$

We now determine the maximum backlog B_{max}. For $T_n < \theta$:

$$B_{max} = (p - R_n)\theta + M + R_n T_n \text{ at } t = \theta \text{ if } p > R_n > r \tag{2}$$

However, in real settings, two cases may arise. One, it may be the case that the maximum network buffer size B_c is smaller than the above mentioned bound B_{max}, in which case, if nothing is done, some traffic may be lost. Moreover, it may also be the case that the above mentioned bound on delay d_{max} is unacceptable for a real-time user who is not prepared to accept a delay, at the network level, larger than a delay constraint d_c. Again, traffic in excess of d_c may be useless to the user and hence lost. Our objective is then to act on the traffic in such a way so as to not exceed the maximum offered buffer size B_c and /or the maximum tolerated delay bound d_c while guaranteeing a loss free performance. This is achieved by the use of a shaper. A shaper shall be introduced between the source and the network (see Figure 6). It has a size B_{sh} which we try to keep as minimal as possible. It has a shaping rate R_{sh} at which the traffic is shaped and sent into the network. A larger value of R_{sh} means a less affected traffic. This is a good feature as the traffic should be minimally altered. An optimal shaper is thus a shaper with minimal buffer size B_{sh} and maximal shaping rate R_{sh}.

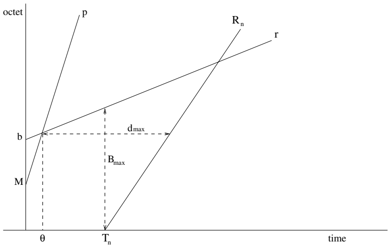


Fig. 5. Arrival Tspec and service curve

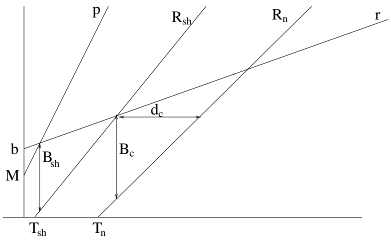


Fig. 6. Shaper between LAN and WAN

4 Shaping

4.1 Regions of Shaping

Adding a shaper to the arriving traffic prior to its entrance to the network is done as follows. A new service curve, corresponding to the actions of the shaper, with parameters (R_{sh},T_{sh}) is set between the arrival curve and the network service curve. This causes the arrival traffic to be first shaped by the newly introduced shaping service curve, the output of which is then sent to the network and served by the network service curve. In what follows, we assume, without loss of generality, that the buffer and server at the network level are fully dedicated to our incoming traffic. Any background traffic shall not interfere with our incoming traffic and shall thus be not explicitly shown, i.e., $T_n = 0$.

Shaping to meet buffer requirement. Let us suppose that the maximum buffer size, B_c, at the network level is smaller than the maximum backlog bound, B_{max}, caused by the non-shaped arriving traffic. The point of introducing a shaper in this case is to assure that the incoming traffic does not exceed B_c for a loss-free network performance. For $\theta' = \frac{b}{R_{sh}-r}$,$B_c = (R_{sh} - R_n)\theta'$ at $t = \theta'$ if $R_{sh} > R_n > r$.
Schematically, and considering the setting of Figure 6, the idea is to vary the shaping curve through the segment indicating B_c. In this case, the shaded region, given in Figure 7, shows the region of shaping. It is wise to note the extreme in this case. It is the shaping curve with shaping rate $h_b < R_n$. This corresponds to a maximal buffer size $B_{sh} > B_{max}$ for the network without shaping. However, R_{sh} is not maximal. Let us note that for R_{sh} more than h_b, the buffer constraint B_c is outperformed uselessly for an even higher shaping rate. The optimal case is given by the shaping curve with shaping rate R_{sh} starting from $T_{sh} = 0$ and the intersection point with $rt + b$ is (θ', y) . This corresponds to maximal R_{sh} or equivalently minimal shaping.

$$R_{sh} = \frac{B_c r - R_n b}{B_c - b} \tag{3}$$

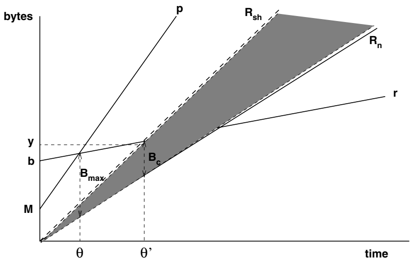


Fig. 7. Region of shaping to meet buffer requirement B_c

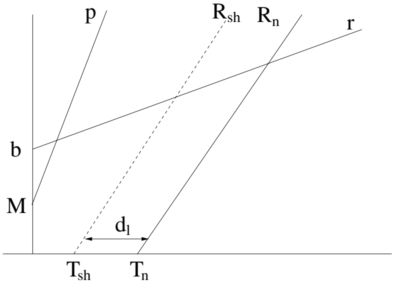


Fig. 8. Region of shaping to meet delay requirement d_c

Again, smaller values of R_{sh} will yield an even smaller B_c uselessly at the cost of higher shaping. Those two cases correspond thus to two feasible shaping parameters depending on the cost of the resources. The first case operates at network buffer less than the target B_c but a high shaping action whereas the second is optimal in view of the shaping action, i.e., large R_{sh}, and network buffer size constraint B_c met.

Shaping to meet delay constraint. In this case, the point of shaping is to reduce the maximum delay to be experienced at the network region from the original d_{max} to a new delay constraint d_c. Let us note that introducing a shaper does not add to the end-to-end delay. The latter shall be just partitioned between the shaper and the network element. This type of partitioning may be useful in an optical context where it is better to hold the packets at the electronic side and not at the optical side where the signal is more prone to being distorted and attenuated. For $\theta' = \frac{b}{R_{sh}-r}$,$d_c = \frac{R_{sh}-R_n}{R_n}\theta'$ at $t = \theta'$ if $R_{sh} > R_n > r$.
This is again achieved by setting appropriate values to R_{sh}. Schematically, the idea is to vary the line the shaping curve through the segment indicating d_c as shown in Figure 8.
The extreme in this case occurs when the shaping curve with shaping rate $h_d < R_n$. This corresponds to a maximal buffer size $B_{sh} > B_{max}$ for the network without shaping. However, R_{sh} is not maximal. Let us note that for R_{sh} more than h_d, the delay constraint d_c is outperformed uselessly for an even higher shaping rate. The optimal case is given by the shaping curve with shaping rate R_{sh} starting from $T_{sh} = 0$ and the intersection point with $rt + b$ is (θ', y).

$$R_{sh} = \frac{R_n(b - rd_c)}{b - R_n d_c} \tag{4}$$

This corresponds to maximal R_{sh} or equivalently minimal shaping. Again, smaller values of R_{sh} will yield an even smaller d_c uselessly at the cost of higher shaping.

4.2 Equation-Based Dynamic Shaping Algorithm

So far, we considered shaping within every interval I_i of length l_N. Our ultimate aim is however to shape the global incoming traffic. Parallel to the idea of partitioning the arrival process so as to locally bound each interval, the shaping scheme introduced in the previous section shall apply to each interval.

It is clear that the shaping rate R_{sh} depends on the arrival curve parameters throughout the whole process. The task in this case is to find optimal, i.e. maximal, shaping rate R_{sh} for each interval I_i such that the buffer constraint and/or delay constraint are satisfied. This is achieved by dynamically changing the shaping rate from one interval to the next. The dynamic shaping algorithm is then as follows.

1. Set observation window size equal to l_N
2. Determine corresponding Tspec in interval $(I_i)_{i=1,\dots,N}$
3. Apply Equations 3 and 4. to set shaping parameters such that
   i. shaping is minimal in the sense of minimal buffer size B_{sh} and maximal shaping rate R_{sh}
   ii. requirements are met, i.e., buffer or delay constraint at network level
   iii. no loss at shaper, i.e. B_{sh} not exceeded.

5 Numerical Results

5.1 Model

We consider the end-to-end system shown in Figure 1. Let the self similar traffic resulting from the LAN sources have the following characteristics: mean = 100 Mbit/s, variance = 10^8, and Hurst parameter H = 0.7.
Let the packets be of maximal size M equal to 1540 bytes. At the network level, let R_n = 227 Mbit/s be the rate of the server, with buffer capacity B_n equal to 100 packets. We assume without loss of generality that a fixed portion of the server at the network level, with service rate R_n and buffer space B_n, is entirely dedicated to our incoming traffic; any background traffic will not be modeled explicitly. This assumption simplifies the analysis and simulation as T_n is equal to zero. In real setting, this amounts to considering a dedicated share of buffer space and service rate.

5.2 Estimation of Arrival Parameters

The first step of our equation-based dynamic shaping algorithm is to estimate the parameters of the incoming traffic into a Tspec format, i.e., peak rate p, mean rate r and maximum burst size b for different observation windows of size l_N. In interval $(I_i)_{i=1,\dots,N}$ of length l_N, as stated in Section 2.1, the peak rate p is equal to the reciprocal of the minimum interarrival time X_{min} and the mean rate r is equal to the reciprocal of the average interarrival times X_{ave}. The complexity lies in the estimation of the maximum burst size b, an essential parameter for a well-defined arrival envelope, performance bounds and shaping issues.

By definition, b corresponds to consecutive arrivals with interarrival times tending to zero as the traffic is observed in a mean phase. In our present work, we observe the consecutive interarrivals of size equal to X_{min} for every interval $(I_i)_{i=1,...,N}$ of length l_N and store the largest value of the corresponding packets in b.

Estimation of mean rate r and peak rate p. Figures 9 and 10 shows the average rate r_i for intervals I_i of lengths equal to 300ms and 1s respectively. We notice that for a given window size l_N, r_i^N varies from one interval to the next keeping the same behavior as the original traffic, i.e., incorporating correlation. On the other hand, the family of $(r_i^N)_{i=1,...,N}$ behaves in the same way in different lengths l_N of intervals $(I_i)_{i=1,...,N}$, i.e., in many time scales. That means the presence of self-similarity property in the sequence $(r_i^N)_{i=1,...,N}$. Figures 9 and 10 shows two times scales $(r_i^N)_{i=1,...,N}$ behaviors : 300ms and 1s. The same remarks remain valid for p.

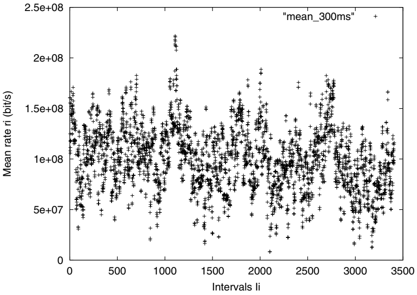


Fig. 9. Average rate r_i during the Interval $I_i : l_N = 300ms$

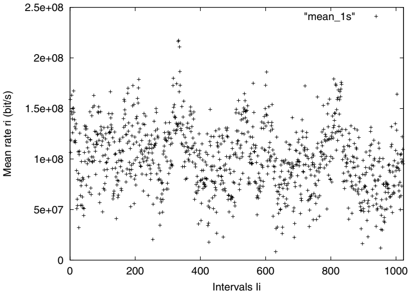


Fig. 10. Average rate r_i during the Interval $I_i : l_N = 1s$

Estimation of burst size b. For each window size l_N, we have observed the interarrival packets during the smallest mean rate r_i. The sum of consecutive interarrivals smaller or equal to the interarrival time within the corresponding peak rate p_i corresponds to the burst size. The obtained values for b vary from 85 packets for small window size l_N to 60 for larger ones.

5.3 Non-constrained Performance

If no shaping is used at the access of the network, Figure 11 shows the probability density function of the queue at the network level. For a no-loss performance, this figure indicates to us that a buffer size of 55 packets is needed at the network server. The maximum delay in the network level in this case is equal to 0.00268 sec.

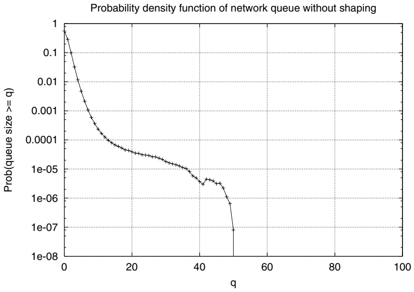


Fig. 11. Network queue PDF without shaping

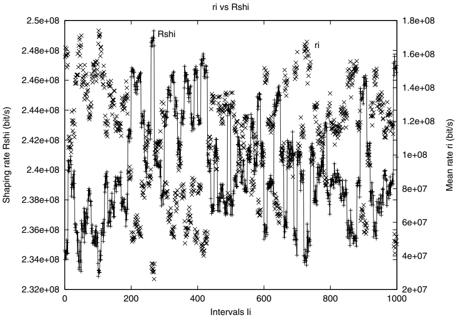


Fig. 12. Average rate r_i vs Shaping rate R_{sh} during I_i

5.4 Buffer-Constrained Performance

In this case, let us assume that in fact, the buffer size B_c at the network level cannot be as large as to hold 55 packets, i.e., if no shaping is used, there will be some loss. Let B_c be large enough to hold 10 packets , a typical buffer size in optical switches. To keep up with a non-loss performance, we need to operate some shaping at the access of the network in order to meet the buffer constraint $B_c = 10$. This brings us to the second step of our algorithm. Based on the arrival Tspec and the service curve equations, we derive the shaping parameters for every interval $(I_i)_{i=1,...,N}$ of length l_N. For intervals of length $l_N = 100$ ms, Figure 12 shows the mean arrival rate r_i versus shaping rate R_{sh} throughout the duration of the connection (100 seconds).

We notice that r_i and R_{sh} are inversely proportional; for every large values of r_i, i.e., high input rate, the value of R_{sh} is small, i.e., a severe shaping is needed to accommodate the buffer constraint and the loss-free performance. The inverse case is also true. The third step of the algorithm is to plug the equation-based shaping parameters back into the simulation model. Figures 13 and 14 show the probability density function of the buffer occupancy at the network level and shaper, respectively, for different observation window lengths l_N. The shaper size is also derived from the equations and the largest value over all intervals is used. This conservatism explains the fact that no loss is observed at the shaper. The independence between the shaping queue PDF's and the interval sizes l_Ns, can be explained by the fact that the shaping rate R_{sh} is adaptive with respect to the incoming traffic in order to meet the non-loss performance. Thus, for each l_N, R_{sh} varies in an inversely proportional way with the mean rate r_i, keeping the shaping queue behavior more or less the same.

The above figures and observation may actually suggest that self-similar traffic, variable at different time scales, may exhibit the same type of variability at those very time scales. If this turns out to be true, it may suggest that observing and monitoring the traffic at small time intervals may be sufficient in constructing and extrapolating its behaviour over larger time scales.

As of the network, we notice that the smallest interval lengths l_N = 65 and 100 ms yield a non-loss performance. This is explained by the fact

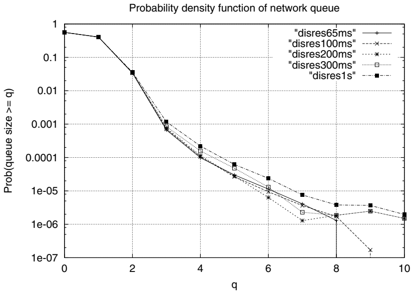


Fig. 13. Network queue PDFs for different lengths Intervals I_i: 65ms, 100ms, 200ms,300ms, 1s

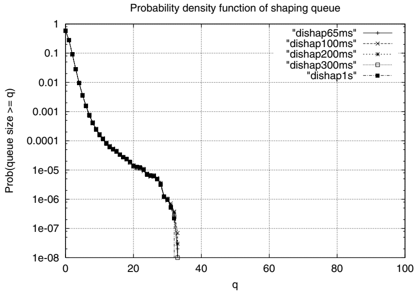


Fig. 14. Shaping buffer PDFs for different lengths Intervals I_i: 65ms, 100ms, 200ms,300ms, 1s

that at those interval lengths, we obtain higher precision for estimation of arrival parameters and hence shaping parameters. For larger interval lengths, $l_N = 200$ and 300 and 1000 ms, some loss, on the order of 2.4 10^{-7}, is observed. This is explained by the fact that for small precision, the arrival parameters are under-estimated. Put in the equations, they yield high shaping rates, or equivalently, soft shaping. This in turn results in loss at the network level.

5.5 Delay-Constrained Performance

Let us assume that in fact, the tolerated maximum delay d_c at the network level cannot be as large as 0.0005 sec , i.e., if no shaping is used, there will be some loss due to the delay being exceeded. Again, to ensure this performance, we need to operate some shaping at the access of the network in order to meet the delay constraint d_c =0.0005 sec. We apply then the three steps of our equation-based algorithm , as done for the buffer-constrained case.

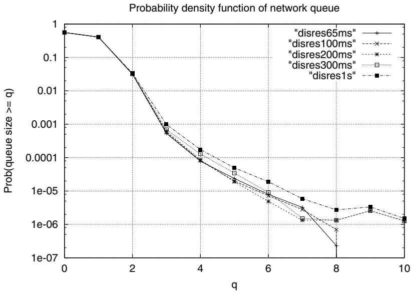


Fig. 15. Network queue PDFs for different lengths Intervals I_i: 65ms, 100ms, 200ms,300ms, 1s

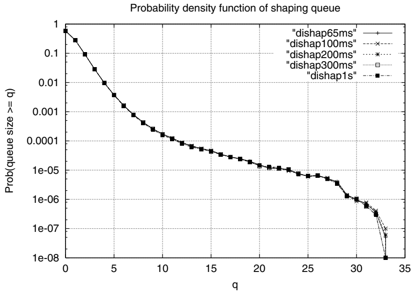


Fig. 16. Shaping buffer PDFs for different lengths Intervals I_i: 65ms, 100ms, 200ms,300ms, 1s

Figures 15 and 16 show the probability density function of the size of the buffer at the network level and shaper, respectively, for different observation window lengths l_N. Table 1 illustrates the maximum delay values obtained by simulation for different window lengths l_N: 65ms, 100ms, 200ms, 300ms and 1s. Again, we notice that the smallest interval lengths $l_N = 65$ and 100 ms yield the target maximum delay.

Table 1. Maximum delay at the network level for different window lengths l_N: 65ms, 100ms, 200ms,300ms, 1s

window lengths l_N	**65ms**	**100ms**	**200ms**	**300ms**	**1s**
maximum delay in the network	0.0004102 s	0.0004219 s	0.0005413 s	0.0005414 s	0.0005414 s

For larger interval lengths, $l_N = 200$ and 300 and 1000 ms, the maximum values of the observed delay exceed the constraint. This is explained by the fact that for small precision, the arrival parameters are under-estimated.

6 Conclusion

In this paper, we focused on self-similar traffic at the input of an optical node. If this traffic is left as is, it cannot satisfy the buffer and/or delay constraint at the network level, which may be very stringent in the optical case. In order to meet those requirements, shaping is essential. In this work, we proposed an equation-based dynamic shaping with three key steps: 1) estimation and fitting of interval-wise incoming traffic into arrival curves, 2) solving into the service curve approach for the shaping parameters in an adaptive manner and 3) fitting the later back into the original model.

As of the first step of our algorithm, we notice that the input estimate reproduces the same self-similar, correlated nature of the original traffic. The shaping parameters derived in step 2 are typically conservative owing to the deterministic nature of the service curve. However, when put back into the original model, i.e., step 3, they are shown to be numerically not very conservative. This may be explained by the correlated nature of the original self-similar traffic.

Future work perspectives shall focus on the following issues. First, the conservatism of the deterministic version of the service curve approach seem to be less apparent in the presence of self-similar, LRD traffic, as shown by the small loss at the network level. It may be wise to quantify to which extent self-similar traffic reduces this conservatism. Second, optimal shaping relies on the trade-off between buffer sizes at the shaper versus network. We intend to tackle this issue by releasing the loss-free determinism at the shaper level where we can in effect tolerate some loss. This can be achieved by more severe shaping action, by decreasing the shaping rate R_{sh}, and hence reaching the buffer size limit. This limit can actually be violated in controlled manner in order to tolerate a loss performance similar to that encountered at subsequent network elements. This feature is desirable and more pragmatic as it is useless to operate a shaping performance too perfect with respect to that of the optical network; after all, what really counts to the user view is the end-to-end performance.

References

1. J-Y. Le Boudec *Application of Network Calculus To Guaranteed Service Networks.* IEEE Trans on Information theory, (44) 3, May 1998.
2. R. L. Cruz *A calculus on network delay, part I: Network elements in isolation.* IEEE Transaction of Information Theory, 37(1):114-121,1991.
3. R. L. Cruz *Quality of Service Guarantees in Virtual Circuit Switched Networks.* IEEE JSAC, 1995.
4. A. Erramili, O. Narayan, and W. Willinger. *Experimental Queuing Analysis with Long-Range Dependence Packet Traffic* IEEE/ACM Transactions on Networking, 4(2), pp. 209-223, Apr. 1996.
5. H. Flower and W. Leland. *Local Area Network Traffic Characteristic, with Implications for Broadband Network Congestion Management* IEEE JSAC, 9(7), pp. 1139-1149, September, 1991.
6. V. Frost and B. Melamed. *Traffic Modeling for Telecommunications Networks* IEEE Communications Magazine, 32(3), pp. 70-80, March, 1994.
7. D. Ferrari and D. Verma. *AA scheme for real-time channel establishement in wide-area networks* IEEE Journal on Selected Areas in Communictions, 8(3):368-379 April, 1990.
8. S. Golestani *Congestion-free transmission of real-time traffic in packet networks* In Proceedings of IEEE INFOCOM'90, pp. 527-142,San Francisco, California, June 1990.
9. C. Kamanek, H. Kanakia and S. Keshav. *Rate controlled servers for very high-speed networks* In Proceedings of IEEE Global Telecommunications Conference, pp. 300.3.1-300.3.9,San Diego, California, December 1990.
10. E. Knightly and H. Zhang. *Traffic characterization and switch utilization usinf deterministic bounding interval dependent traffic models* In Proceedings of IEEE INFOCOM'95,Boston, MA, April 1995.
11. W. Leland, M. Taqqu, W. Willinger and D. Wilson. *On the Self-Similar Nature of Ethernet Traffic (Extended Version)* IEEE/ACM Transactions on Networking, 2(1), pp. 1-15, February 1994.
12. V. Paxon and S. Floyd. *Wide-Area Traffic: The failure of Poisson Modeling* IEEE/ACM Transactions on Networking, 3(3), pp; 226-244, June 1995.

Is Admission-Controlled Traffic Self-Similar?*

Giuseppe Bianchi, Vincenzo Mancuso, and Giovanni Neglia

Università di Palermo, Dipartimento di Ingegneria Elettrica
Viale delle Scienze, 90128 Palermo, Italy
bianchi@elet.polimi.it, {vincenzo.mancuso,giovanni.neglia}@tti.unipa.it

Abstract. It is widely recognized that the maximum number of heavy-tailed flows that can be admitted to a network link, while meeting QoS targets, can be much lower than in the case of markovian flows. In fact, the superposition of heavy-tailed flows shows long range dependence (self-similarity), which has a detrimental impact on network performance. In this paper, we show that long range dependence is significantly reduced when traffic is controlled by a Measurement-Based Admission Control (MBAC) algorithm. Our results appear to suggest that MBAC is a value added tool to improve performance in the presence of self-similar traffic, rather than a mere approximation for traditional (parameter-based) admission control schemes.

1 Introduction

The experimental evidence that packet network traffic shows self-similarity[1] was first given in [1], where a thorough statistical study of large Ethernet traffic traces was carried out. This paper stimulated the research community to explore the various taste of self-similarity. This phenomenon has been observed in wide area Internet traffic and many causes that contribute to self-similarity for both TCP and UDP traffic aggregates have been now more fully understood [2,3,4,5].

In this paper, we focus our attention on traffic generated by sources non-reactive to network congestion (e.g. real-time multimedia streams). The traffic aggregate offered to a network link results from the superposition of several individual flows. It has been proven [6] that self-similarity or Long Range Dependence (LRD) arises when individual flows have heavy-tailed[2] periods of activity/inactivity. This result is valid asymptotically as the number of sources increases.

We are interested in the practical implications of self-similarity on the design of Call Admission Control (CAC) schemes. In this paper we assume, for convenience, a traffic scenario composed of homogeneous flows. In these conditions, a

* This research is supported by European Community and MIUR in the frame of the Pollens project (ITEA, if00011a).

[1] In this paper we use the terms self-similarity and long range dependence in an interchangeable fashion, because we refer to asymptotic second order self-similarity (for details [7]).

[2] A random variable is said to be "heavy-tailed" when its cumulative distribution function converges to $F(t) \sim 1 - at^{-c}$, as $t \to \infty$ with $1 < c < 2$, being a a constant.

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 327–339, 2002.

traditional (parameter-based) CAC rule simply checks that the number of admitted flows never exceeds a maximum threshold N_t. This threshold is selected so that target Quality of Service (QoS) requirements (e.g. loss ratio, delay percentiles, etc.) are met. In what follows we refer to this CAC scheme as MAXC (Maximum number of Calls).

A large amount of work (see [7]) has shown that self-similarity has a detrimental impact on network performance. For the same link capacity and buffer size scenario, the Quality of Service (i.e. loss/delay performance) experienced by LRD traffic results worse than that experienced by Short Range Dependent (SRD) traffic, e.g. modelled as Markov processes. The straightforward interpretation of these results, in terms of traditional CAC, is that self-similarity is a key factor which reduces the maximum number N_t of flows that can be admitted.

We argue that the above interpretation is questionable, as it does not account for recent progress in admission control schemes, and specifically the emergence and increasing popularity of Measurement-Based Admission Control (MBAC) approaches [8,9,10,11]. Unlike traditional CAC methods, which rely on a-priori knowledge of the statistical characterization of the offered traffic, MBAC algorithms base the decision whether to accept or reject an incoming call on run-time measurements on the traffic aggregate process.

The aim of this paper is to present results which show that MBAC approaches appear capable of smoothing the self-similarity of the accepted traffic aggregate. In this sense, MBAC approaches are not merely "approximations" of ideal CAC schemes in situations where the statistical traffic source characterization is not fully known. On the contrary, this paper shows that MBAC schemes are an effective and important way to cope with the high variability of LRD traffic, and their adoption leads to significant performance advantages with respect to traditional CAC schemes (refer to [11] for an initial insight on the performance advantages of MBAC in an LRD traffic scenario).

The rest of the paper is organized as follows. In section 2 we briefly describe the MBAC principles and we discuss the important role of MBAC in the presence of self-similar traffic. The specific MBAC algorithm adopted and the methods to evaluate self-similarity are described in section 3. Numerical results are presented and discussed in section 4. Finally, concluding remarks are given in section 5.

2 Measurement Based Admission Control

It is frequently assumed that the ultimate MBAC goal is to reach the "ideal" performance of a parameter-based CAC scheme. In fact, MBAC schemes are traditionally meant to approximate the operation of a parameter-based CAC. They cannot rely on the detailed a-priori knowledge of the statistical traffic characteristics, as this information is not easy supplied in an appropriate and useful form by the network customer. Therefore, their admission control decisions are based on an estimate of the network load obtained via a measurement process that runs on the accepted traffic aggregate.

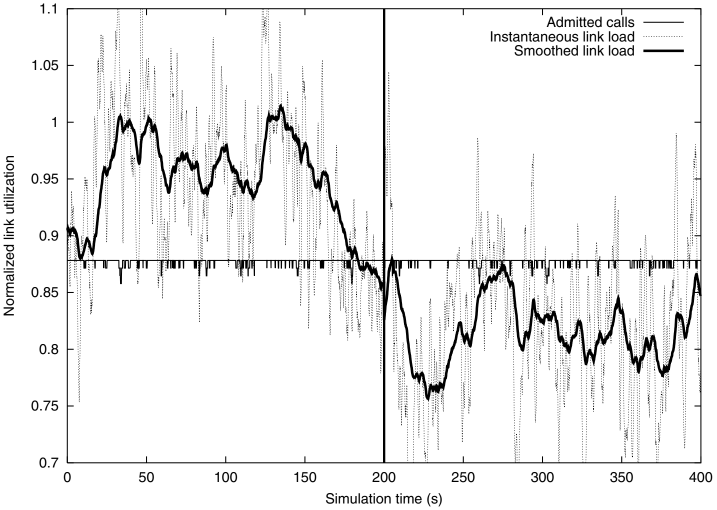


Fig. 1. Traditional Admission Control operation

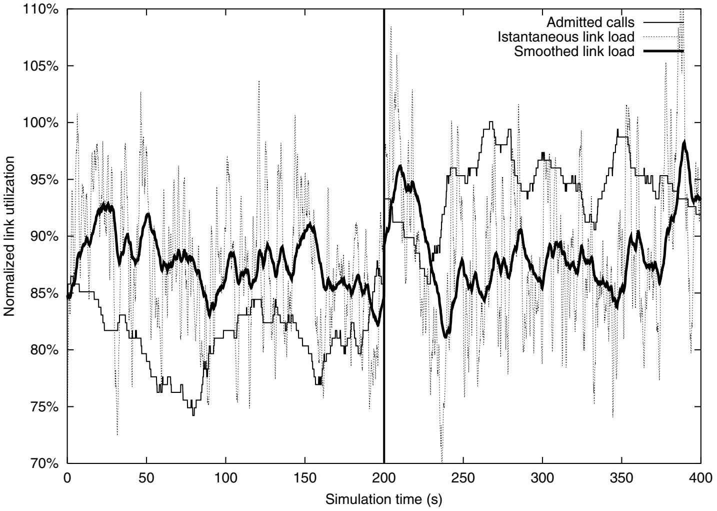


Fig. 2. Measurement-Based Admission Control operation

However, a closer look at the basic principles underlying MBAC suggests that, in particular traffic conditions, these schemes might outperform traditional parameter-based CAC approaches. An initial insight into the performance benefits of MBAC versus parameter-based algorithms in an LRD traffic scenario is given in [11]. In this paper, we present additional results that confirm the superiority of MBAC and we justify them showing that MBAC algorithms are able to reduce the self-similarity of the traffic aggregate generated by the admitted heavy-tailed sources. In other words, we argue that MBAC schemes are not just "approximations" of parameter-based CAC, but they are *in principle superior* to traditional CAC schemes when self-similarity comes into play.

An intuitive justification can be drawn by looking at the simulation traces presented in figures 1 and 2 (simulation details are described in section 3). Each figure reports two selected 200 s simulation samples, which for convenience have been placed adjacently. The y-axis represents the normalized link utilization. The figures report: i) the number of accommodated calls normalized with respect to the link capacity; ii) the instantaneous link load, for graphical convenience averaged over a 1 s time window, and iii) the smoothed link load, as measured by the autoregressive filter adopted in the MBAC, whose time constant is of the order of 10 seconds.

Figure 1 reports results for a parameter-based CAC scheme (MAXC). According to this scheme, a new flow is accepted only if the number of already admitted flows is lower than a maximum threshold N_t. In the simulation run N_t was set to 129, which corresponds to a target link utilization of about 88%, and a very high offered load (650%) was adopted. As a consequence, the number of flows admitted to the link sticks, in practice, to the upper limit.

The leftmost 200 simulation seconds, represented in Figure 1, show that, owing to LRD of the accepted traffic, the load offered by the admitted sources is consistently well above the nominal average load. Traffic bursts even greater than the link capacity are very frequent. On the other hand, as shown by the rightmost 200 seconds, there are long periods of time in which the system remains under-utilized. The criticality of self-similarity lies in the fact that the described situation occurs at time scales, e.g. the one shown in the figure, which dramatically affect the loss/delay performance.

A very different situation occurs for MBAC schemes. Figure 2 reports results for the simple MBAC scheme described in section 3.2. In this case, new calls are blocked as long as the offered-load measurement is higher than 89% (the values 129 in MAXC and 89% in MBAC were selected so that the resulting average throughputs were the same). In this case, we see from both leftmost and rightmost plots that the offered-load measurement fluctuates slightly around the threshold. However, long term traffic bursts are dynamically compensated by a significant decrease of the number of admitted calls (leftmost plot). The opposite situation occurs when the admitted calls persistently emit under their nominal average rate: indeed the rightmost plot shows that in these periods the number of admitted calls significantly increases. This "compensation" capability of MBAC schemes leads us to conclude that MBAC is well-suited to operating in LRD traffic conditions: the quantitative analysis carried out in section 4, in fact, confirms this insight.

3 The Simulation Scenario

To obtain simulation results, we have developed a C++ event-driven simulator. A batch simulation approach was adopted. The simulation time is divided into

101 intervals, each lasting 300 simulated minutes, and results collected in the first "warm-up" time interval are discarded.

As in many other admission control works [10,11], the network model consists of a single bottleneck link. The reason is that the basic performance aspects of MBAC are most easily revealed in this simple network configuration rather than in a multi-link scenario. The link capacity was set equal to 2 Mbps, and an infinite buffer size was considered. Thus, QoS is characterized by the delay (average and 99th delay percentiles) experienced by data packets rather than packet loss as in [11]. The rationale for using delay instead of loss is threefold. Firstly, loss performance depends on the buffer size adopted in the simulation runs, while delay performance does not require a choice of buffer size (we have actually used infinite buffer size). Secondly, the loss performance magnitude may be easily inferred, for a given buffer size, from the analysis of the distribution of the delay, which can be well summarized via selected delay percentiles. Thirdly, and most importantly, a limited buffer size acts as a smoothing mechanism for traffic bursts. Large packet losses, occurring during severe and persistent traffic bursts (as that expected for self-similar traffic), have a beneficial congestion control effect on the system performance. Conversely, in a very large buffer scenario, the system is forced to keep memory of non-smoothed traffic bursts and therefore performance is further degraded in the presence of high traffic variability[3].

As our performance figures, we evaluated link utilization (throughput) and delay distribution, summarized, for convenience of presentation, by the average and 99th delay percentile. The 95% confidence intervals have been evaluated. In all cases, throughput results show a confidence interval always lower than 0.3%. Instead, despite the very long simulation time, higher confidence intervals occur for 99th delay percentile results: less than 5% for MBAC results, and as much as 25% for MAXC results (this is an obvious consequence of the self-similarity of the MAXC traffic aggregate).

3.1 Traffic Sources

For simplicity, we have considered a scenario composed of homogeneous flows. Each traffic source is modelled as an ON/OFF source. While in the ON state, a source transmits 1000 bit fixed size packets at a Peak Constant Rate (PCR) randomly generated in the small interval 31 to 33 Kbps (to avoid source synchronization effects at the packet level). Conversely, while in the OFF state, it remains idle. The mean value of the ON and OFF periods were set, respectively,

[3] Specifically, this justifies the very different performance results we obtain in high utilization conditions when compared with the loss-utilization performance frontier presented in [11] for LRD sources. In that paper, unlike our results presented in figure 4, it appears that performance of MBAC schemes tend to converge to the performance of traditional CAC schemes - i.e. the MAXC algorithm - as the utilization increases. A theoretical justification for this behavior can be found in [16], where the authors derive a formula to estimate the "correlation horizon" (which results to scale in linear proportion to the buffer size), beyond which the impact on loss performance of the correlation in the arrival process becomes nil.

equal to 1 s and 1.35 s (Brady model for voice traffic). This results in an average source rate $r = 0.4255 \cdot E[PCR] \approx 13.6$ Kbps. ON and OFF periods were drawn from two Pareto distributions with the same shaping parameter $c = 1.5$ (so they exhibit heavy-tails).

Simulation experiments were obtained in a dynamic scenario consisting of randomly arriving flows. Each flow requests service from the network, and the decision whether to admit or reject the flow is taken by the specific simulated admission control algorithm. A rejected flow departs from the network without sending any data, and does not retry its service request again. The duration of an accepted flow is taken from a lognormal distribution [12] with mean 300 s and standard deviation 676 s (we adopted unitary variance for the corresponding normal distribution as reported in [12]), but call duration is extended to the end of the last ON or OFF period. Because of this, the real call-lifetime exhibits longer mean (320 s) and infinite variance. If the last burst were cut off, the process variance would become finite.

The flow arrival process is Poisson with arrival rate λ calls per second. For convenience, we refer to the normalized offered load $\rho = \lambda \cdot r \cdot T_{hold}/C_{link}$, being r the mean source rate, T_{hold} the average call duration and C_{link} the link capacity.

3.2 Measurement-Based Admission Control Algorithm

Rather than using complex MBAC proposals, we have implemented a very basic MBAC approach. The rationale for the choice of a very simple MBAC scheme is twofold. Firstly, it has been shown [11] that different MBAC schemes behave very similarly in terms of throughput/loss performance. It appears that the length of the averaging periods and the way in which new flows are taken into account, are much more important than the specific admission criteria. Secondly, and more importantly, our goal is to show that the introduction of measurement in the admission control decision is the key to obtain performance advantages versus the MAXC approach, rather than the careful design of the MBAC algorithm. In this perspective the simpler the MBAC scheme is, the more general the conclusions are.

The specific MBAC implementation is described as follows. A discrete time scale is adopted, with sample time $T = 100$ ms. Let $X(k)$ be the load, in bits/sec, entering the link buffer during the time slot k, and let $B(k)$ be a running bandwidth estimate, smoothed by a simple first order autoregressive filter

$$B(k) = \alpha B(k-1) + (1-\alpha)X(k)$$

We chose $\alpha = 0.99$, corresponding to about 10 s time constant in the filter memory.

Consider now a call requesting admission during the slot $k+1$. The call is admitted if the estimated bandwidth $B(k)$ is less than a predetermined percentage of the link bandwidth. By tuning this percentage, performance figures can be obtained for various accepted load conditions.

An additional well-known issue in MBAC algorithm design [9] is that, when a new flow is admitted, the slow responsiveness of the load estimate will not immediately reflect the presence of the new flow. A solution to prevent this performance-impairing situation is to artificially increase the load estimate to account for the new flow. In our implementation, the actual bandwidth estimate $B(k)$ is updated by adding the average rate of the flow (i.e. $B(k) := B(k) + r$).

3.3 Statistical Analysis of Self-Similarity

The Hurst parameter H is able to quantify the self-similarity of the accepted traffic aggregate. For a wide range of stochastic processes $H = 0.5$ corresponds to uncorrelated observations, $H > 0.5$ to LRD processes and $H < 0.5$ to SRD processes.

In order to evaluate H, we used the well known three methods described below. All methods receive in input a realization $X(i)$ of the discrete-time stochastic process representing the load offered, during a 100 ms time window, to the link buffer by the accepted traffic aggregate. The methods adopted are:

1. **Aggregate Variance** [13]. The original series $X(i)$ is divided into blocks of size m and the aggregated series $X^{(m)}(k)$ is calculated as

$$X^{(m)}(k) = \frac{1}{m} \sum_{i=(k-1)m+1}^{km} X(i) \qquad k = 1, 2, \ldots$$

   The sample variance of $X^{(m)}(k)$ is an estimator of $Var\left(X^{(m)}\right)$; asymptotically:

$$Var\left(X^{(m)}\right) \sim \frac{Var(X)}{m^{2(1-H)}}$$

2. **Rescaled Adjusted Range (R/S)** [13]. For a time series $X(i)$, with partial sum $Y(n) = \sum_{i=1}^{n} X(i)$, and sample variance $S^2(n)$, the R/S statistics or the rescaled adjusted range, is given by:

$$\frac{R}{S}(n) = \frac{1}{S(n)} \left[\max_{0 \le p \le n} \left(Y(p) - \frac{p}{n} Y(n) \right) - \min_{0 \le p \le n} \left(Y(p) - \frac{p}{n} Y(n) \right) \right]$$

   Asymptotically:

$$E\left\{ \frac{R}{S}(n) \right\} \sim C n^H$$

3. **Wavelet Estimator** [14] (see [15] for a freely distributed Matlab implementation). We recall that the spectrum of an LRD process $X(t)$ exhibits power-law divergence at the origin $W_X(f) \sim c_f |f|^{(1-2H)}$. The method recovers the power-law exponent $1 - 2H$ and the coefficient c_f turning to account the following relation

$$E\left\{ d_X^2(j, l) \right\} = 2^{j(1-2H)} c_f C$$

   where $d_X(j, l) =< X, \psi_{l,j} >$ are the coefficients of the discrete wavelet transform of the signal $X(t)$, i.e. its projections on the basis functions $\psi_{l,j}$, constructed by the mother wavelet through scaling and translation (2^j and l are respectively the scaling and the translation factor).

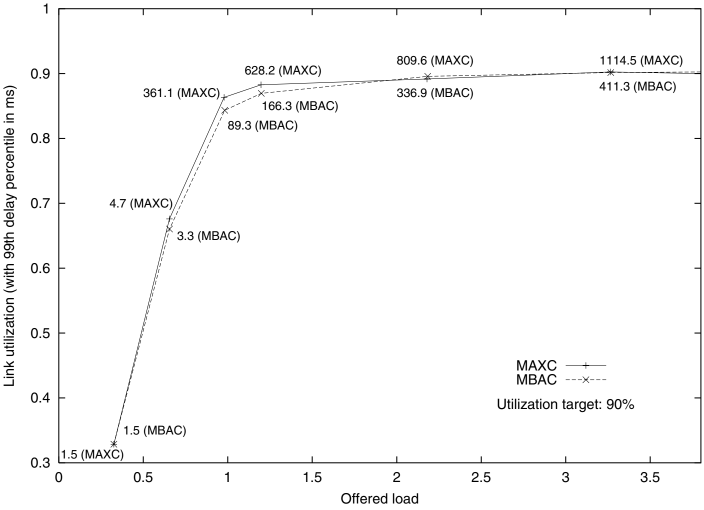


Fig. 3. Link utilization vs offered load

A common problem is to determine over which scales LRD property exists, or equivalently the alignment region in the logscale diagrams. Using the fit test of the matlab tool [15] we determined for our traces the range from 2000 s -11th octave- to 250000 s -18th octave- (the two last octaves were discarded because there were too few values). All the three methods were applied over this scale.

4 Performance Evaluation

A problem arising in the comparison of different CAC schemes is the definition of a throughput/performance operational trade-off. In general, CAC schemes have some tunable parameters that allow the network operator to set a suitable *utilization target* and a consequent QoS provisioning. For example, in the case of the ideal MAXC algorithm, a higher setting of the threshold value results in an increased system throughput, at the expense of delay performance. By adjusting these parameters, CAC rules can be designed to be more aggressive or conservative with regard to the number of flows admitted.

Results presented in figure 3 were obtained by setting the MAXC and MBAC tuning parameters so that a target 90% link-utilization performance is achieved in overload conditions. The figure compares the throughput/delay performance (99th delay percentiles, measured in ms, are numerically reported) of MBAC and MAXC, versus the normalized offered load. Minor differences can be noted in the capability of the considered schemes to achieve the performance target (as expected, MAXC converges faster than MBAC to the utilization target). A much more interesting result is the significantly lower MBAC 99th delay performance versus the MAXC one.

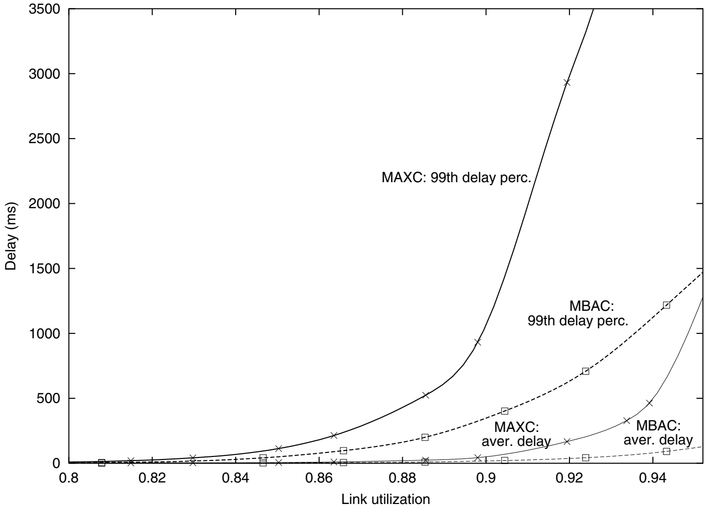


Fig. 4. Delay performance vs link utilization

It is restrictive to limit the investigation to a single level of performance, but it is preferable to compare different CAC schemes for a wide range of link utilization targets (and, correspondingly, QoS performance), obtained by varying the CAC threshold parameters. Unless otherwise specified, all results presented in what follows are obtained in large overload conditions (650% offered load).

Rather than varying the offered load, figure 4 compares MBAC and MAXC by plotting their QoS performance versus the link utilization (following [11], the QoS versus utilization curve is called *Performance Frontier*). Specifically, the figure reports the delay/utilization performance frontiers of MAXC and MBAC. Both average and 99th delay percentiles are compared. The figure shows that the performance improvement provided by MBAC is remarkable, especially for large link utilization.

We argue that the performance enhancement of MBAC over MAXC is due to the beneficial effect of MBAC in reducing the self-similarity of the accepted traffic aggregate. To quantify this statement, tables 1 and 2 report the Hurst-parameter estimates obtained with the three methods described in section 3.3, along with the corresponding CAC settings (maximum call number for MAXC; link utilization threshold for MBAC), and the achieved link utilization[4]. We see that the methods provide congruent estimates. Results are impressive, and show

[4] As we said, these results were obtained with an offered load equal to 650%. It may be remarked that the different results of MBAC and MAXC in shaping traffic reduce in lighter load conditions, and vanish for very low offered loads (when neither MBAC nor MAXC enforce call rejections). By the way, in this situation, traffic self-similarity is irrelevant in terms of performance, as traffic QoS requirements are met. Additional results, not presented here due to space constraints, show that MBAC capability to reduce self-similarity becomes evident as soon as the offered load approaches the

Table 1. Hurst-parameter estimate for MAXC controlled traffic

MAXC				
Thresh (calls)	Thrput %	Hurst Variance	Hurst R/S	Hurst Wavelet
105	71.8	0.73	0.79	0.78
110	74.7	0.77	0.78	0.76
115	78.3	0.74	0.78	0.80
120	81.5	0.73	0.79	0.76
125	84.5	0.71	0.79	0.75
127	86.8	0.77	0.77	0.75
130	88.7	0.78	0.76	0.75
132	90.3	0.68	0.73	0.75
135	91.7	0.72	0.72	0.77
137	93.4	0.71	0.77	0.76
140	94.7	0.78	0.80	0.74

Table 2. Hurst-parameter estimate for MBAC controlled traffic

MBAC				
Thresh (util%)	Thrput %	Hurst variance	Hurst R/S	Hurst Wavelet
70	69.1	0.55	0.48	0.55
74	73.0	0.60	0.55	0.57
78	76.9	0.58	0.54	0.58
82	80.8	0.56	0.50	0.58
86	84.6	0.55	0.51	0.60
88	86.6	0.52	0.49	0.53
90	88.5	0.60	0.52	0.57
92	90.4	0.54	0.52	0.56
94	92.4	0.51	0.46	0.56
96	94.3	0.58	0.52	0.58
98	96.2	0.58	0.53	0.57

that the Hurst parameter decreases from about 0.75, in the case of MAXC, to about 0.5 for MBAC. It is interesting to note that 0.75 is the Hurst-parameter value theoretically calculated in [6] when a flow has heavy-tailed periods of activity/inactivity with a shaping parameter $c = 1.5$ (the formula is $H = (3 - c)/2$). In conclusion, table 2 quantitatively supports our thesis that self-similarity is a marginal phenomenon for MBAC controlled traffic (the achieved Hurst parameter is very close to 0.5, which represents SRD traffic).

To quantify the time behavior of the two MAXC and MBAC traffic-aggregate time series, figure 5 reports a log-log plot of the aggregate variance, computed as described in section 3.3. While the two curves exhibit similar behavior for small values of the aggregation scale, the asymptotic slope of the MAXC plot is very different from the MBAC one. We recall that the asymptotic slope β is related to the Hurst parameter by $\beta = 2H - 2$. The lines corresponding to $H = 0.50$, $H = 0.55$, $H = 0.75$ and $H = 0.80$ are plotted in the figure as reference comparison. Note that the figure 5 appears to suggest that the MBAC-controlled traffic is not self-similar (Hurst parameter close to 0.5).

Similar considerations can be drawn, with greater evidence, by looking at figure 6, which reports a log-log plot of the estimated squared wavelet coefficients $d_x^2(j,l)$ versus the basis-function time scale. The figure shows that, for large time scales, the MBAC-controlled traffic plot tends to lay on a horizontal line (the asymptotic slope γ is related to the Hurst parameter by $\gamma = 1 - 2H$, and thus a horizontal line corresponds to $H = 0.50$, the $H = 0.80$ case is also plotted in the figure as reference comparison).

Finally, figures 5 and 6 show that the MBAC curve departs from the MAXC curve at a time scale of the order of about 100 seconds. Although a thorough

target utilization threshold, and Hurst-parameter values reach those presented in table 2 as soon as the offered load becomes 10-20% greater than this target.

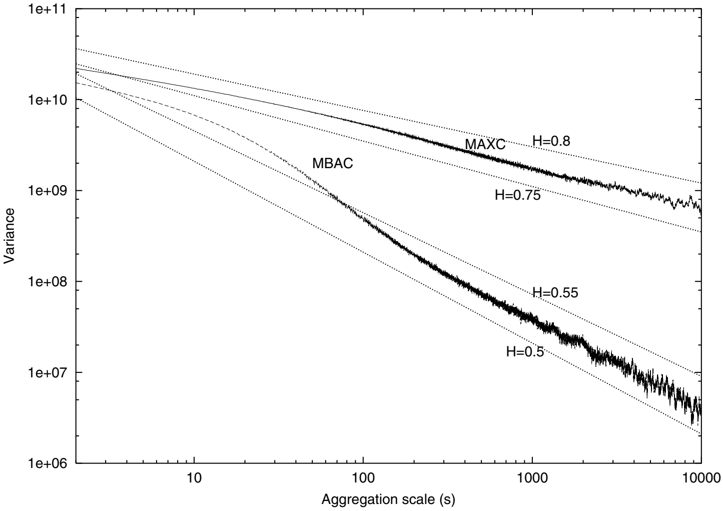


Fig. 5. Aggregated variance plot

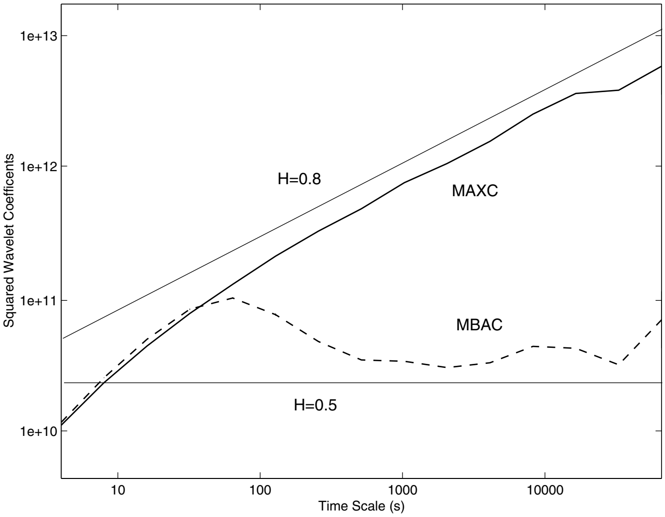


Fig. 6. Wavelet coefficients plot

understanding of the emergence of such a specific time scale is outside the scope of the present paper, we suggest that it might have a close relationship with the concept of “critical time scale” outlined in [10].

5 Conclusions

The results presented in this paper appear to suggest that the traffic aggregate resulting from the superposition of Measurement-Based Admission Controlled flows shows a very marginal long range dependence. This is not the case for traffic controlled by a traditional parameter-based admission control scheme.

We feel that there are two important practical implications of our study. Firstly, our study support the thesis that MBAC is not just an approximation of traditional CAC schemes, useful when the statistical pattern of the offered traffic is uncertain. On the contrary, we view MBAC as a value-added traffic engineering tool that allows a significant increase in network performance when offered traffic shows long range dependence. Secondly, provided that the network is ultimately expected to offer an admission control function, which we recommend should be implemented via MBAC, our results seem to question the practical significance of long range dependence, the widespread usage of self-similar models in traffic engineering, and the consequent network oversizing.

Acknowledgements. The authors want to thank the graduate student Vito Imburgia for his collaboration in the software development.

References

1. W.Leland, M.Taqqu, W.Willinger, D.Wilson, “On the Self-Similar Nature of Ethernet Traffic”, Trans. on Networking, Vol. 2, No. 1, pp. 1-15, Feb. 1994.
2. J.Beran, R.Sherman, W.Willinger, and M.S.Taqqu, “Variable-bit-rate video traffic and long-range dependence”, IEEE Transactions on Communications, 1995
3. K.Park, G.T.Kim, M.E.Crovella, “On the Relationship Between File Sizes, Transport Protocols, and Self-Similar Network Traffic”, Proceedings of the International Conference on Network Protocols, pp. 171-180, October, 1996.
4. M.E.Crovella and A.Bestavros, “Self-Similarity in World Wide Web Traffic: Evidence and Possible Causes”, Trans. on Networking, 5(6):835–846, Dec. 1997.
5. A.Feldmann, A.C.Gilbert, W.Willinger, “Data network as cascades: Investigating the multifractal nature of the Internet WAN Traffic”, Computer Communications Review 28 (1998)
6. W.Willinger, M.Taqqu, R.Sherman, D.Wilson, “Self-Similarity Through High-Variability: Statistical Analysis of Ethernet LAN Traffic at the Source Level”, Trans. on Networking, Vol. 5, No. 1, pp. 71-86, Feb. 1997.
7. [edited by] K.Park and W.Willinger “Self-Similar Network Traffic and Performance Evaluation”, Wiley Inter-Science, 2000
8. R.Gibbens, F.P.Kelly, “Measurement-Based Connection Admission Control”, Proc. of 15th International Teletraffic Congress, June 1997
9. S.Jamin, P.B.Danzig, S.Shenker, L.A.Zhang, “A measurement-based admission control algorithm for integrated services packet networks”, Trans. on Networking Vol. 5, No. 1, Feb 1997, pp. 56-70.
10. M.Grossglauser, D.Tse, “A Framework for Robust Measurement Based Admission Control”, Trans. on Networking Vol. 7, No. 3, July 1999.

11. L.Breslau, S.Jamin, S.Shenker, “Comments on the Performance of Measurement Based Admission Control Algorithms”, Proc. of IEEE Infocom 2000, Tel Aviv, Israel, March 2000.
12. V.Bolotin, “Modeling Call Holding Time Distributions for CCS Network Design and Performance Analysis”, IEEE Journal on Selected Areas in Communications 12, 3 (Apr. 1994), 433–438.
13. J.Beran, “Statistics for long-memory processes”, Chapman & Hall, 1994
14. P.Abry, D.Veitch, “Wavelet Analysis of Long-Range Dependent Traffic”, IEEE Transactions on Information Theory, 44(1), pp. 2-15, January 1998.
15. `http://www.emulab.ee.mu.oz.au/~darryl/secondorder_code.html`
16. M.Grossglauser and J.Bolot, “On the Relevance of Long Range Dependence in Network Traffic”, Trans. on Networking, 1998.

Analysis of CMPP Approach in Modeling Broadband Traffic

R.G. Garroppo, S. Giordano, S. Lucetti, and M. Pagano

Department of Information Engineering, University of Pisa
Via Diotisalvi 2 - 56126 Pisa - Italy
{r.garroppo, s.giordano, s.lucetti, m.pagano}@iet.unipi.it

Abstract. The CMPP (Circulant Modulated Poisson Process) modeling approach represents an appealing solution since it provides the integration of traffic measurement and modeling. At the same time, it maintains the Markovian hypothesis that permits analytical transient and steady-state analyses of queueing systems using efficient algorithms. These relevant features of CMPP approach has driven us to analyze in more details the fitting procedure when it is applied to actual broadband traffic. In the paper, investigating the estimation algorithm of model parameters, we emphasize the difficulty of CMPP in capturing the upper tail of marginal distribution of actual data, which leads to an optimistic evaluation of network performance. As shown in the paper, a simple relation exists between the number of significant eigenvalues obtained by the spectral decomposition and the peak rate that the CMPP structure is able to capture. The relation evidences the difficulties of CMPP to model actual traffic, characterized by long tailed distribution, as well as traffic data with the well accepted hypothesis of gaussian marginal.

1. Introduction

The CMPP approach for modeling arrivals process by means of a circulant modulated Poisson process, provides a technique for integration of traffic measurement and modeling [10], maintaining, at the same time, the Markovian hypothesis that permits analytical transient and steady-state studies of queueing systems using efficient algorithms [9]. The developed modeling theory has permitted to study the impact of power spectrum, bispectrum, trispectrum, and marginal distribution of the input process on queueing behavior and loss rate. These studies have highlighted the key role played on the queueing performance by the marginal distribution, especially in the low frequencies region [8]. The technique for the construction of a CMPP that matches marginal distribution and autocorrelation function of the observed process has been presented in [2,9], where the authors showed simulation results with measured traffic data to prove the goodness of this approach. In this paper, further analysis of CMPP fitting procedure will be presented, highlighting a limitation of the mentioned algorithm in matching accuracy for the marginal distribution of observed rate process. Moreover, the presented study determines the maximum peak rate captured by the CMPP model once the spectrum has been matched and emphasizes the necessity of a

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 340-351, 2002.

CMPP structure containing a large number of effective eigenvalues to adequately capture even the light tail of a gaussian function, usually accepted as realistic for traffic distribution in the core network [1]. The relevance of these considerations is related to the impact of the marginal distribution tail of input traffic on queueing behaviour. Indeed, as shown in the numerical analysis Section, optimistic performance are estimated when the peak rate is not matched. On the other hand, the actual traffic rate has a marginal distribution that in some cases exhibits a tail heavier than Gaussian [3,5]; under such condition, the CMPP models result inadequate to estimate realistic queueing performance. Lastly, some advice to overcome the exposed limitation are briefly introduced.

2. Background on CMPP

The fitting procedure of a CMPP model mainly consists of three steps [9], which are briefly summarized in this section. In the first step, the autocorrelation function of the observed rate process is estimated and then matched by a sum of exponentials (with complex parameters λ_k) weighted by real and strictly positive power coefficients ψ_k. This matching is a non-linear problem and cannot be solved directly. An approximate, but quite accurate, solution is obtained by using the Prony algorithm [6] to express the autocorrelation function in terms of complex exponentials with complex coefficients, and then satisfying the constraints on the ψ_k's (which must be real and strictly positive) by matching the power spectral density (PSD) using the nonnegative least square (NNLS) method.

The second step aims to design the transition frequencies matrix $\underline{\underline{Q}}$ of the underlying modulant continuous time Markov chain. In order to fit the PSD of the modeled process, the eigenvalues of $\underline{\underline{Q}}$ must contain all the λ_k's obtained in the previous step; the use of a circulant matrix permits to solve the inverse eigenvalues problem. An efficient procedure to solve this problem is the Index Search Algorithm (ISA), presented in [1].

The last step is then the estimation of a vector $\underline{\gamma}$ associated to the Poissonian generation of arrivals in each state of the modulating Markov chain, such that the model matches the cumulative distribution function (CDF), *F(x),* of the observed rate process. In more details, the fitting procedure starts considering that the autocorrelation function of a CMPP model with *N* states is expressed by the following:

$$R_{CMPP}(\tau) = \psi_0 + \sum_{l=1}^{N-1} \psi_l \cdot \exp(\lambda_l \cdot |\tau|) \qquad \textbf{(2.1)}$$

with positive real ψ_l's. The Fourier transform of (2.1) can be expressed by:

$$S_{CMPP}(\omega) = 2 \cdot \pi \cdot \psi_0 \cdot \delta(\omega) + \sum_{l=1}^{N-1} \psi_l \cdot b_l(\omega) \qquad \textbf{(2.2)}$$

where $b_l(\omega) = \mathrm{F}\left[\exp(\lambda_l \cdot |\tau|)\right] = \dfrac{-2 \cdot \lambda_l}{\omega^2 + \lambda_l{}^2}$, and $\int_{-\infty}^{\infty} b_l(\omega) d\omega = 1$; hence, the ψ_l's represent the power associated to each λ_l. The λ_l's are the eigenvalues of the transition matrix, which must include all the "effective" ones that derive by the exponential decomposition of *R(τ)*, the autocorrelation function of the measured rate process. Using the Prony method, the estimated *R(τ)* can be written as

$$R(\tau) \cong \sum_{k=0}^{p} \psi_{P,k} \cdot \exp(\lambda_{P,k} \cdot |\tau|) \ . \tag{2.3}$$

The presence of a constant term in R_{CMPP} *(τ)* requires $\lambda_{P,0}$ to be imposed equal to zero, and consequently $\psi_0 = \psi_{P,0}$: this is simply obtained applying the Prony method to the autocovariance function $C(\tau) = R(\tau) - \bar{\gamma}^2$ ($\bar{\gamma}$ is the mean value of the observed traffic rate), since $R(\tau \to \infty) \to \bar{\gamma}^2$, and from (2.3) $\psi_{P,0} = \ \bar{\gamma}^2$. After the NNLS matching, the expression (2.3) remains substantially unchanged and can be rewritten as

$$R(\tau) \cong \psi_{P,0} + \sum_{k=1}^{p} \psi_{P,k} \cdot \exp(\lambda_{P,k} \cdot |\tau|) \tag{2.4}$$

being aware that *p*, the $\psi_{P,k}$'s and the $\lambda_{P,k}$'s may not be the same as those of (2.3) (they surely will not be in the case of complex eigenvalues). The order *p* of the exponential decomposition may be much less than the order *N* of the model, and thus in the construction of the transition matrix only few λ_l's will be imposed equal to the $\lambda_{P,k}$'s. Indicating with $\underline{i}$ the vector of indices (of dimension *p*) such that $\lambda_{i[k]} = \lambda_{P,k}$, the relation $\psi_{i[k]} = \psi_{P,k}$ consequently holds. On the other hand, in order to obtain $R_{CMPP}(\tau) \cong R(\tau)$, all the other ψ_l's will be imposed equal to zero.

After having determined the transition matrix $\underline{\underline{Q}}$ (note that many solutions are possible for each set of eigenvalues, since the order *N* of the matrix is higher than the number *p* of desired eigenvalues), the third step, i.e. the design of the rate vector $\underline{\gamma}$ such that $F_{CMPP}(x) \cong F(x)$, involves the minimisation of the distance between $F_{CMPP}(x)$ and $F(x)$, which is obtained by using the Nelder-Mead Simplex Search method. Since $F_{CMPP}(x)$ is a piecewise step function, which jumps by *1/N* at each value γ_i in $\underline{\gamma}$, the task is to determine the optimal vector $\underline{\gamma}$ which minimises the quantity

$$\sum_{i=0}^{N-1} |\gamma'_i - \gamma_i| \tag{2.5}$$

where $\underline{\gamma}'$ is obtained by the quantization of $F(x)$ in levels, whose amplitude is *1/N*.

Defining $\beta_i = \sqrt{\psi_i} \cdot \exp(j\vartheta_i)$, i=0,1,…,N-1, the vector $\underline{\beta} = [\beta_0, \beta_1, ..., \beta_{N-1}]$ represents the Discrete Fourier Transform of $\underline{\gamma}$, and its Inverse can be expanded as

$$\gamma_i = \bar{\gamma} + \sum_{l=1}^{N-1} \sqrt{\psi_l} \cdot \exp\left[-j\left(\frac{2 \cdot \pi \cdot i \cdot l}{N} - \vartheta_l\right)\right] \text{, for i=0,1,2,…,N-1}$$

where the expression of β_l has been substituted.

In order to obtain real γ_i, $\underline{\beta}$ must exhibit the Hermitian property (i.e. $\beta_{N-l} = \beta_l^*$, which corresponds to $\psi_{N-l} = \psi_l$ and $\vartheta_{N-l} = -\vartheta_l$). Indeed, if $\underline{\beta}$ does not satisfy the Hermitian property, its Inverse Finite Fourier Transform $\underline{\gamma}$ cannot be real. Under the condition of Hermitianity on $\underline{\beta}$, the above relation assumes the following expression

$$\gamma_i = \bar{\gamma} + \sum_{l=1}^{N-1} \sqrt{\psi_l} \cdot \cos\left(\frac{2 \cdot \pi \cdot i \cdot l}{N} - \vartheta_l\right) \text{, for i=0,1,2,…,N-1} \qquad \textbf{(2.6)}$$

that permits to estimate $\underline{\gamma}$ by applying the Nelder-Meade Simplex Search method to (2.5) as a function of $\underline{\vartheta}$. The Hermitian conditions on β_l are automatically satisfied for those power coefficients related to conjugated complex pairs of eigenvalues, but cannot stand for real ones, since only one ψ_l is associated to each of them. To overcome this problem, each real eigenvalue needs to be considered twice. In order to maintain the same correlation structure (or equivalently the same PSD), the corresponding power coefficients will be assumed equal to half of the original ψ_l's.

3. Analysis of Fitting Procedure

The investigation presented in this work involves the last step of the fitting procedure and evidences a relevant limitation on the tail behavior of the marginal distribution of CMPP models. This limitation may considerably affect the evaluation of queueing performance of actual traffic, leading to an underestimation of network resources needed to guarantee the target QoS expressed in terms of loss probability.

The first observation on the fitting procedure is that, putting $\tau=0$ in (2.4), the variance σ^2 of the rate process can be expressed as the sum of the ψ_l for $l=1, 2, ..., N-1$. On the other hand, the maximum theoretical rate achievable by the model is derived by (2.6) putting all the cosines equal to +1. In this case, a second relation involving ψ_l's can be simply obtained:

$$\gamma_{MAX} - \bar{\gamma} = \sum_{l=1}^{N-1} \sqrt{\psi_l} \ . \qquad \textbf{(3.1)}$$

The maximum rate deviation from the mean value is then limited by (3.1), under the constraint $\sum_{l=1}^{N-1} \psi_l = \sigma^2$. As we stated before, only the p ψ_l's associated to the effective eigenvalues are non zero. Among these p power coefficients, some are related to real λ_l, hence each of them needs to be split into two terms with halved magnitude. Therefore, the resulting set of couples (λ_l, ψ_l) after this operation consists of q elements, with $p \le q << N$. In the remaining of the paper, we will refer to (λ_l, ψ_l) as elements of this set; consequently if $\lambda_l \in \Re \Rightarrow \exists\ \lambda_{N-l} = \lambda_l$ and $\psi_{N-l} = \psi_l$, with ψ_l equal to half of the original power coefficient obtained by NNLS algorithm. Thus relation (3.1) can be rewritten as

$$\gamma_{MAX} - \bar{\gamma} = \sum_{k=1}^{q} \sqrt{\psi_{i[k]}} \qquad \textbf{(3.2)}$$

where $\underline{i}$ is now a vector of indices of dimension q.

In most actual cases, the measured peak rate is quite higher than the mean value and then, in order to capture the long tailed behavior of the rate distribution, the sum in equation (3.2) should be as large as possible. The problem of maximizing (3.2) with the constraint $\sum_{k=1}^{q} \psi_{i[k]} = \sigma^2$ can be easily solved using the Lagrange-Multipliers method, leading to the solution

$$\psi_{i[k]} = \sigma^2 / q \ . \qquad \textbf{(3.3)}$$

Consequently

$$\max_{\underline{\psi}} \{\gamma_{MAX} - \bar{\gamma}\} = \sum_{k=1}^{q} \sqrt{\psi_{i[k]}} \cong \sqrt{q} \cdot \sigma \qquad \textbf{(3.4)}$$

holds.

This equation represents the intrinsic limitation of CMPP in terms of maximum achievable rate as a function of the number q of effective eigenvalues, in the hypothesis of evenly distributed power coefficients. Considering that the distribution of the amplitudes of the dominant ψ_i's will hardly be like (3.3), the γ_{MAX} value obtained from the above relation represents only an upper bound, actually difficult to reach. However, (3.4) gives an indication on the minimum number of exponentials required to reach a target peak rate, fixed the variance and the mean value of the observed rate process. As an example, suppose that the observed process presents a gaussian marginal distribution with mean $\bar{x}$ and variance σ^2; hence, the CDF is

$$F(x) = 1 - Q\left(\frac{x - \bar{x}}{\sigma}\right) \text{ where } Q(y) = \frac{1}{2\pi}\int_y^{\infty} \exp\left(-\frac{x^2}{2}\right) dx \qquad \textbf{(3.5)}$$

Using a 500 state CMPP to model this rate process, the CDF results divided in intervals whose heights are 1/500=2·10^{-3} (the high number of states has been chosen in order to obtain a fine quantization of CDF). The maximum level of the quantized CDF is then limited to the value 0.998, corresponding to $(x - \bar{x})/\sigma$ equal to 3.09. The comparison of this relation with (3.4) leads to q=3.09^2≅9.55. Therefore, the original autocovariance should be decomposed in, at least, 10 exponentials; more likely they will not be sufficient, since the assumption that all power coefficients ψ_i's are of equal magnitude is not easily verified. Indeed, a more realistic scenario is that few ψ_i's (around 6 or 8, as supported by the analysis in the Numerical Section) will be dominant with respect to the others and consequently the reproduced peak rate will be such that $\gamma_{MAX} - \bar{\gamma} = \sqrt{6} \cdot \sigma$, in correspondence of whom the original CDF will assume the value $1 - Q\left(\sqrt{6}\right) \cong 0.993$. Hence, the tail of the model will be shorter than the one of the observed rate process. The relevance of this drawback can be pointed out envisaging that, especially for traffic whose power spectrum is concentrated in the lower region of the frequencies [7] (this assumption is supported by the self similar nature highlighted by the recent modeling results based on the analysis of acquired traffic data [11]), the tail of the marginal distribution has a deep impact on the network resources required to guarantee a target loss probability.

Note that from (2.6) $\bar{\gamma} - \gamma_{MIN} = \sum_{k=1}^{q} \sqrt{\psi_{i[k]}}$ can also be derived, which, together with (3.2), implies the following general relation

$$\left|\gamma - \bar{\gamma}\right|_{MAX} = \sum_{k=1}^{q} \sqrt{\psi_{i[k]}} \; . \qquad \textbf{(3.6)}$$

4. Numerical Results

To test the relevance of the presented analysis, we consider two sets of simulations: the first one is carried out applying the CMPP fitting procedure on synthetic data with a well defined and known spectral decomposition, whereas the second one refers to actual traffic data. In the first simulation scenario, we have generated two traces having the same mean value, variance and power spectrum, differing only by their probability density functions (one is gaussian and the other one is triangular). The values of the mean and the variance of the data ($\bar{\gamma}$ =4500 cells per second, σ^2=10^6 cps^2) have been chosen in order to obtain a negligible probability of having negative values in the gaussian rate trace. In order to build the CMPP models of the two traces, we have first decomposed the autocovariance functions into a sum of complex exponentials. Then, applying the NNLS algorithm, only the three couples of eigenvalues shown in Table 1.(a) (here and in the remaining of the paper, the "**" reminds that the real eigenvalue is considered twice and that the corresponding ψ has been already halved in magnitude, according to the procedure described in the previous sections) have turned out to be associated to power coefficients ψ different from zero (note that $\sum_i \psi_i$ equals the imposed variance σ^2).

Table 1. Eigenvalues with non-zero power coefficient deriving from the NNLS algorithm: (*a*) Synthetic Gaussian trace; (*b*) trace "Videoconference"; (*c*) trace "October89"

ψ [cps^2]	λ [rad/sec]
1.72 e+5	*-0.78±j 6.92*
2.62 e+5	*-1.26±j 3.00*
6.6e+4	*-0.14 ***

(*a*)

ψ [cps^2]	λ [rad/sec]
9.72 e+4	*-2.85±j 6.57*
1.25 e+5	*-5.92 ***
6.40 e+5	*-0.628 ***
2.28 e+6	*-0.315 ***

(*b*)

ψ [cps^2]	λ [rad/sec]
1.46 e+5	*-1.23±j6.03*
2.74 e+6	*-2.35±j 0.554*
5.25 e+5	*-0.033 ***

(*c*)

According to (3.6), the maximum deviation achievable using this set of eigenvalues is equal to

$$|\gamma - \bar{\gamma}|_{MAX} = \sum \sqrt{\psi} \cong 2367 \text{ cps} . \tag{4.1}$$

The obtained value is quite close to the limit $\sqrt{6} \cdot \sigma \cong 2450$ cps, since the power coefficient magnitudes have a quasi-uniform distribution.

Two CMPP models have been built from these eigenvalues to match the two different CDFs. The γ_{MAX}'s for the two models have resulted equal to 6660 cps and 6500 cps for the triangular and the gaussian hypothesis, respectively. These limits do not consider the exponential and I.I.D. generation of arrivals (Poissonian microdynamics) in each state of the CMPP model, which affects the upper tails of marginal distributions, as well as the peak rates of generated traces. In particular, the latter ones are higher than the respective γ_{MAX}'s, (see the third column of Table 2, which summarizes the main statistics of the analyzed data traffic and of the related traces generated by the corresponding CMPP models). The results of the fittings in terms of PSD and complementary probability (CP) of the generated traces in the gaussian hypothesis are shown in Fig. 1. The excellent matching of the PSD is not accompanied by an equivalently good fitting of the distribution function. This mismatch is not easily revealed by the CDF's comparison, hence a CP plot is always required to appreciate possible differences in the upper tails. In the triangular case, we observed good fitting results for both PSD and CP plots, but we do not report the relative figures here for sake of simplicity.

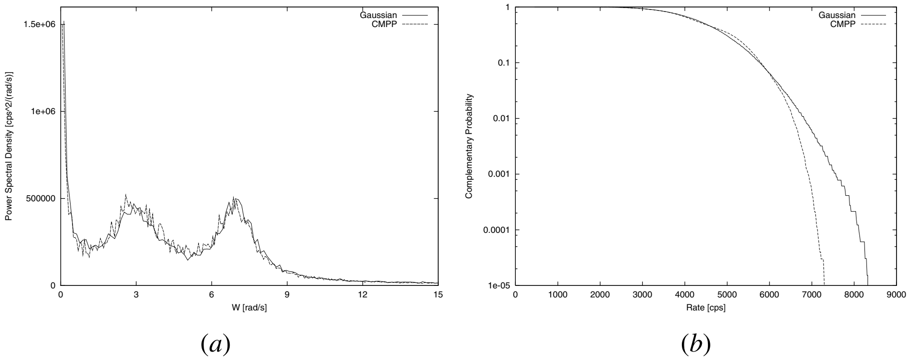
In order to estimate the errors in the evaluation of queueing performance introduced by the mismatching of the CDF tail, we have analyzed the results obtained by means of discrete-event simulations. The simulations have been carried out feeding a FCFS G/D/1/K queueing system with the four traces; in the remaining of the paper we will indicate with μ the servant constant cell rate. The size K of the buffer has been fixed equal to 250 cells, corresponding to a maximum delay introduced by the queue, variable with the normalized offered load, around 50msec. The traffic is completely contained in the LF and the MF region of the queue [7]. Therefore, no further filtering of traffic is possible without losing a portion of its spectrum, i.e. MF components, which would affect the queueing performance evaluation. The results are shown in the last two columns of Table 4 and emphasize as the reduced peak of the trace related to the gaussian case leads to a slightly optimistic resources allocation (nearly 6%). It is important to highlight that the CMPP designs have been repeated using different values for N, ranging from 200 to 500, but no appreciable difference has been noticed.

In the second set of simulations, two real traffic traces have been considered: the two data sets are related to a videoconference service and a LAN traffic trace. The description of the characteristics of the videoconference traffic data are described in [4], whereas the second trace is the well known "October89" trace collected at Bellcore Labs. In both analyses, the data (i.e. the estimates of the rate processes) refer to the equivalent number of ATM cells (calculated as the number of transmitted bytes divided by 48) per second observed in non overlapped time intervals of length Tu. In the first case, we assume Tu equal to the frame period, i.e. 40 ms, whereas for the second traffic trace the value of 200 ms has been chosen.

The particular shape of the autocovariance function of the videoconference trace has led to a spectral decomposition characterized by the eigenvalues reported in Table 1.(*b*). In this real scenario, we have obtained only few eigenvalues with non-zero power coefficients, enforcing the hypothesis introduced in the previous section regarding the number of dominant ψ's. In this case the maximum achievable deviation is equal to $\left|\gamma-\bar{\gamma}\right|_{MAX}=\sum\sqrt{\psi}\cong 5950$ cps and the mean value is approximately 4350 cps. Hence, we should expect a CMPP peak rate of about 10300 cps, against a measured peak rate of about 21000 cps. The peak rate error is very high (nearly 43%) due to the very unfavorable condition on the power coefficient distribution. Indeed, the power spectral decomposition of the considered trace presents only a single couple of dominant effective real eigenvalues and, at the same time, its marginal distribution exhibits an upper tail behavior heavier than gaussian [4]. The modeling results are shown in Fig. 2.(*a*) and 2.(*b*), which represent the PSD and the complementary probability matching, respectively. In particular, Fig. 2.(*b*) shows clearly that the obtained CMPP peak rate is much lower than the observed 21000 cps, as expected. This large difference in terms of peak rate leads to a very optimistic evaluation of the queueing behavior, as pointed out by the simulations results, see Table 2, columns 4 and 5, which contain the servant cell rate required in order to reach a cell loss probability of 10^{-4} and 10^{-5}.

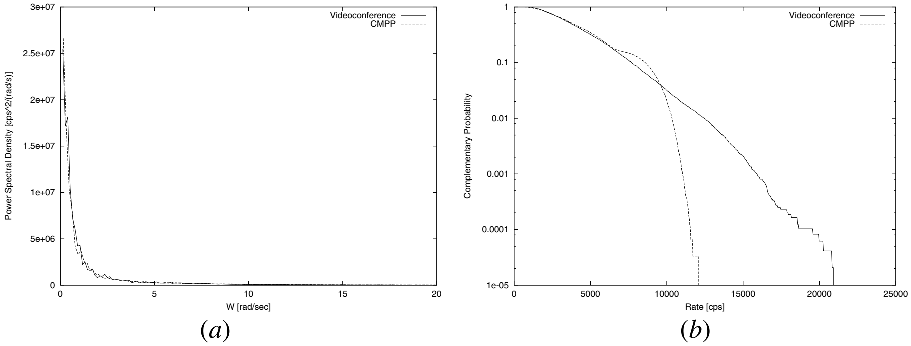
The last analysis refers to the first 1000 s of the above mentioned LAN traffic trace; we do not consider the entire data set since a shift of the mean value has been noticed out of this time period. The relevant eigenvalues obtained after the NNLS matching of the spectrum are reported in Table 1.(*c*), and correspond to a theoretical maximum deviation $\left|\gamma-\bar{\gamma}\right|_{MAX}\cong 5524$ cps. Adding the estimated mean (about 6016 cps) to this value, the limit 11540 cps is obtained, with respect to a model $\gamma_{MAX}=\max_{i}(\gamma_i)$ of 11145 cps and of a data set peak of 17800 cps. Fig. 3.(*a*) and 3.(*b*) present the matching of PSD and complementary probability respectively: the former shows the good fitting of the considered second order statistic, whereas the latter confirms the limitation of the model in capturing the upper tail behavior of marginal distribution.

The last two columns of Table 2 clearly evidence the entity of the relative error in resources allocation when the CMPP models does not capture the upper tail of marginal distribution (and particularly the peak cell rate).



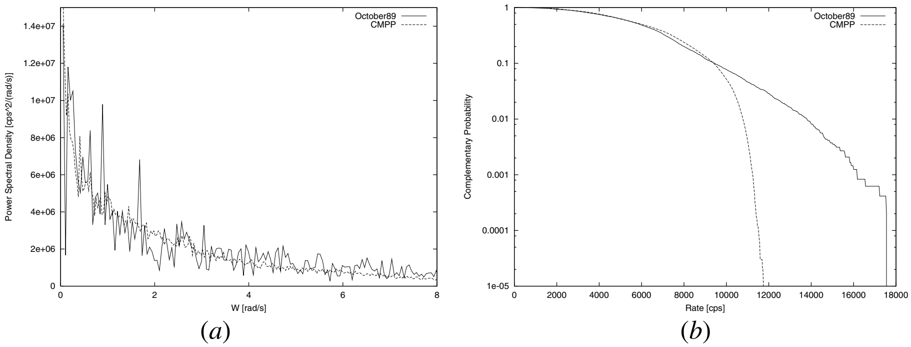
(*a*) (*b*)

Fig. 1. Comparison of (*a*) Power Spectral Density and (*b*) Complementary Probability of Synthetic Traffic Data – Gaussian Case



(*a*) (*b*)

Fig. 2. Comparison of (*a*) Power Spectral Density and (*b*) Complementary Probability of "Videoconference" Trace



(*a*) (*b*)

Fig. 3. Comparison of (*a*) Power Spectral Density and (*b*) Complementary Probability of "October89" Trace

Table 2. Relevant statistics of analysed traces and corresponding CMPP models

	Mean [cps]	Variance [cps^2]	Peak [cps]	μ (P_{loss}=10^{-4}) [cps]	μ (P_{loss}=10^{-5}) [cps]
Triangular (CMPP model)	4493 (4506)	1.00e+6 (1.00e+6)	7000 (7500)	5930 6025 (+1.6%)	6033 6170 (+2.3%)
Gaussian (CMPP model)	4490 (4503)	1.00e+6 (0.99e+6)	8344 (7330)	6110 5855 (-4.2%)	6390 6010 (-5.9%)
October89 (CMPP model)	6016 (6030)	7.48e+6 (6.03e+6)	17600 (11780)	15360 10330 (-32.7%)	16120 10590 (-34.3%)
Videoconference (CMPP model)	4350 (4410)	6.15e+6 (6.10e+6)	20975 (12100)	16400 9670 (-41.0%)	18520 9820 (-47.0%)

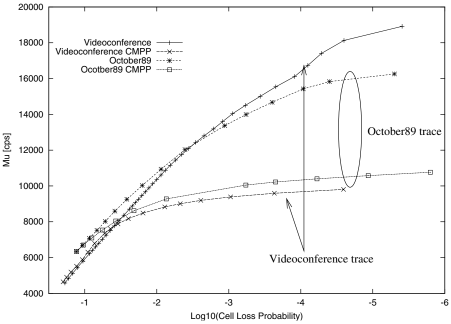


Fig. 4. Performance Comparison

In the cases of the videoconference and LAN traces, a more complete analysis of modeling performance is shown in Figure 4, where the target CLP is plotted versus the servant cell rate needed to guarantee it. The analysis of Table 2 points out two important results: the first one concerns the critical behavior of the CMPP model even in the gaussian case, which implies an optimistic resource allocation (4 to 6%). The second result highlights as actual traffic exhibits a slower decay of marginal distribution with respect to the gaussian hypothesis leading to a less suitable environment for the CMPP approach, evidenced by the large errors suffered (in terms of peak rate, matching of the upper tail of marginal distribution and consequently of the network resources needed to guarantee a target cell loss probability).

5. Improvement Proposals to Overcome the CMPP Drawback

We have observed that, under the same conditions on q, the uniform distribution of ψ's magnitudes is the only one that guarantees the maximum model peak γ_{MAX}. Thus, it is desirable to have a spectral decomposition with power coefficients exhibiting this feature. To the aim of coming close to the uniform distribution and increasing the parameter q, a straight solution can be to split each dominant exponential in the sum of two or more terms. The j-th term of the exponential decomposition is represented by the parameters λ_j and ψ_j , as described in Section 2. Our suggestion is to obtain an equivalent contribution to the autocorrelation function using a number B of power coefficients (associated to the given exponential). This procedure results in having B terms $\lambda_{j,i}, \psi_{j,i}$ for each couple λ_j, ψ_j, where $\lambda_{j,i} = \lambda_j$ and $\psi_{j,i} = \psi_j / B$, *i=1,2,...,B.*

Consequently, the autocorrelation function remains unchanged, whereas the contribution of the decomposed couple λ_j, ψ_j to the peak rate becomes $\sum_{i=1}^{B} \sqrt{\psi_{j,i}} = \sum_{i=1}^{B} \sqrt{\frac{\psi_j}{B}} = \sqrt{B} \cdot \sqrt{\psi_j}$, i.e. $\sqrt{B}$ times the original one (i.e. $\sqrt{\psi_j}$).

Unfortunately, this solution also presents a drawback, whose relevance needs further investigation: the increased number of eigenvalues to be assigned to $\underline{\underline{Q}}$ can cause the increment of the minimum number of the model states N that permits the solution of the ISA problem. As a consequence, an higher N means that the model parameters derivation takes a longer time, thus limiting the use of CMPP approach in a real time performance estimator.

Another possible approach to overcome the presented limitation is to increase the value of T. In such a way, the peaks of the traffic data will be reduced, hence making easier to capture them by the CMPP model. Unfortunately, also this procedure presents some drawbacks. First of all, a limit exists on the maximum value of the time quantum T. Indeed, using higher T can cause the loss of a significant portion of information associated to the traffic data [7], leading to inaccurate queueing performance evaluation. Furthermore, increasing T reduces the variance σ^2 of the resulting trace. Consequently, the constraint on the ψ_i's sum ($\sum_i \psi_i = \sigma^2$) produces a set of power coefficients of reduced magnitude, vanishing the advantage gained by the trace peak reduction. Further study is then needed to highlight the trend of the peak rate decay with respect to the time quantum T in different simulation scenarios, and to derive a relation with the tail behavior.

6. Conclusions

The paper presents an analysis of the algorithm for the measurement-based parameters estimation of CMPP models, raising some warnings to be aware of in the use of this modeling approach. In particular, the main result of the paper is the analytical derivation of an intrinsic limitation of the fitting procedure in the modeling of traffic characterized by long-tailed marginal distribution. Furthermore, the analysis shows that even in the gaussian hypothesis, a CMPP structure containing a large number of effective eigenvalues is necessary to adequately fit the CDF.

In general, the limitation manifests when the difference between the peak and the mean rate of the traffic data set exceeds few times its standard deviation. In this condition the CMPP model cannot match with sufficient accuracy the behavior of the CDF upper tail, leading to optimistic prediction of the network resources needed to guarantee the target QoS (in terms of cell loss probability).

Discrete-event simulations driven by actual traffic data and synthetic traces, generated according to the corresponding CMPP models, have confirmed the results of the proposed analytical study. Moreover, the analysis of the simulation results shows the practical relevance of the CMPP limitation in a single server queueing system, a relevant case study in performance comparison. In particular, the analyzed cases highlight the optimistic resources allocation produced by the fitting errors on the CDF. The presented limitation reduces the field of applicability of the CMPP modeling approach, hence further studies are needed to overcome this drawback. To this aim, some advice are presented as possible solutions to be investigated.

References

1. R. G. Addie, M. Zuckerman, T.D. Neame, "Broadband Traffic Modeling: Simple Solutions to Hard Problems" IEEE Communications Magazine, August 1998, pp.88-95
2. H. Che, San-qi Li, "Fast Algorithms for Measurement-Based Traffic Modeling", IEEE Journal on Selected Areas in Communications, June 1998, pp. 612-625
3. R. G. Garroppo, S. Giordano, S. Porcarelli, G. Procissi, "Testing α-stable processes in modelling broadband teletraffic" Proc. of IEEE ICC 2000, New Orleans, Louisiana, USA, 18-22 June, 2000
4. R. G. Garroppo, S. Giordano, M. Pagano, "Stochastic Features of VBR Video Traffic and Queueing Working Conditions: a Simulation Study using Chaotic Map Generator" in Proc. of IFIP Broadband Communications '99, Hong Kong, November 1999
5. A. Karasaridis, D. Hatzinakos, "A Non-Gaussian Self-Similarity Processes for Broadband Heavy-Traffic Modeling", in Proc. of GLOBECOM 98, pp. 2995-3000, Sidney, 1998
6. Steven M. Kay, "Modern Spectral Estimation: Theory & Application", Prentice-Hall, 1988.
7. Y. Kim, San-qi Li, "Timescales of Interest in Traffic Measurement for Link Bandwidth Allocation Design", Proc. IEEE, Infocom '96, March 1996, pp. 738-748
8. San-qi Li, Chia-Lin Hwang, "Queue Response to Input Correlation Functions: Continuous Spectral Analysis", ACM/IEEE Transactions on Networking, December 1993, pp. 678-691
9. San-qi Li, Chia-Lin Hwang, "On the Convergence of Traffic Measurement and Queueing Analysis: A Statistical-Matching and Queueing (SMAQ) Tool", IEEE/ACM Transactions on Networking, February 1997, pp. 95-110
10. San-qi Li, S. Park, D. Arifler, "SMAQ: A Measurement-Based Tool for Traffic Modeling and Queueing Analysis. Part I: Design Methodologies and Software Architecture", IEEE Communications Magazine, August 1998, pp. 56-65
11. W.Willinger, M.S. Taqqu, A. Erramilli, "A bibliographical guide to self-similar traffic and performance modelling for modern high speed networks", in F.P. Kelly, S. Zachary and I. Ziedins eds., Stochastic networks: Theory and Applications in Telecommunication Networks, Vol. 4 of Royal Statistical Society Lecture Notes Series, pp. 91-104, Oxford University Press, Oxford, 1996

A Mathematical Model for IP over ATM

Irena Atov and Richard J. Harris

Royal Melbourne Institute of Technology,
GPO Box 2476V Melbourne, Vic. 3001, Australia,
{Irena, Richard}@catt.rmit.edu.au
http://www.catt.rmit.edu.au/

Abstract. We consider IP over ATM networking scenario, widely deployed today, where Asynchronous Transfer Mode (ATM) is used as a backbone network to provide high-speed transport for the Internet Protocol (IP) traffic. Telecommunication network carriers and Internet Service Providers (ISPs), that have deployed ATM as their backbone networks, need to model and characterize the IP traffic in order to plan and manage these networks to meet specified performance measures demanded by their customers. This paper addresses the problem of modelling IP traffic that is being transported over links of an ATM network. It provides mathematical models that can be used to take measurement data and translate it into a form that is suitable for various planning and management functions performed by network carriers.

1 Introduction

We consider an IP over ATM networking scenario (Fig. 1), where ATM is used as a backbone network to provide high-speed transport for the IP internetwork traffic. The IP over ATM networking scenario involves setting up a mesh of permanent virtual circuits (PVCs) between Internet Gateway Routers (IGRs) around an ATM cloud, and the Next Hop Resolution Protocol (NHRP) achieves a similar result with switched virtual circuits (SVCs). In such a scenario, the Internet traffic is first aggregated by IGRs before being sent on an ATM backbone. Typically, in this case each ATM virtual connection carries the traffic corresponding to a potentially large number of IP connections. We address the problem of modelling the aggregate traffic of multiple IP connections as it enters the ATM backbone into a form that is suitable for the dimensioning process of the ATM network.

ATM networks use the notion of *equivalent bandwidth* to transform the multilevel traffic problem associated with the ATM networks into a multi-slot circuit-switched problem, which enables ATM networks to be dimensioned in a fashion similar to telephony networks [1] [2] [3]. The Guérin et al [4] method for equivalent bandwidth assumes that the cell generation of each traffic type in the ATM network is a fluid flow on-off process, characterized by the parameters: mean cell rate, peak cell rate, and mean burst period. The ATM Forum has defined the same set of parameters as source traffic descriptors for the ATM Variable Bit

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 352–363, 2002.

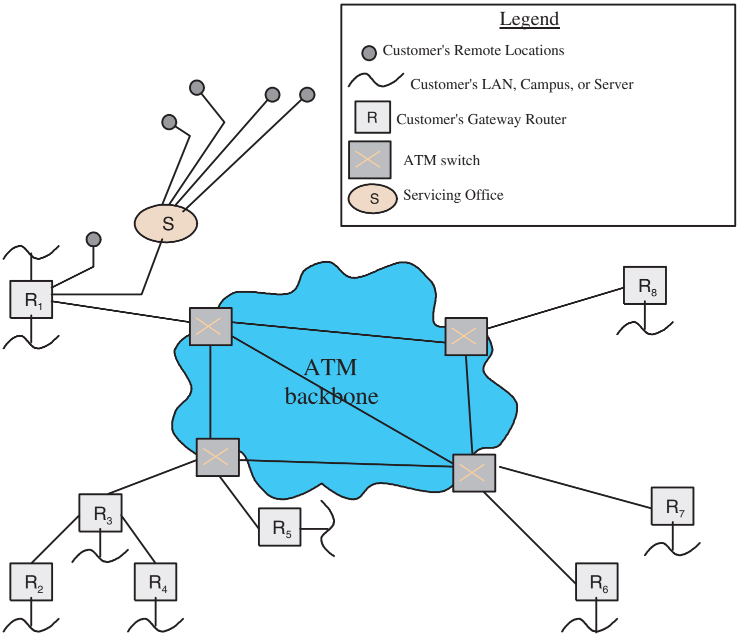

Legend
Customer's Remote Locations
Customer's LAN, Campus, or Server
R
Customer's Gateway Router
ATM switch
S
Servicing Office
S
R1
R2
R3
R4
R5
R6
R7
R8
ATM
backbone

Fig. 1. IP Over ATM Networking Scenario

Rate (VBR) service [8]. For ATM networks, VBR is one of the effective solutions to accommodate IP traffic and in order to guarantee high quality service, it is important to select appropriate VBR parameter values. In addition, these three parameters are used for traffic policing at the ingress of the ATM network [8].

Our contribution can be summarized as follows. First, we model the traffic from multiple IP connections that is being transported over links of ATM network as an aggregate traffic stream characterized by the on-off traffic parameters: mean cell rate, peak cell rate, and mean burst period. The motivation is clearly to obtain a traffic model that is consistent with that used in [4] for determining an equivalent bandwidth. Then, we discuss approaches for incorporating our model into the ATM dimensioning procedures.

For the modelling, we analyze separately Transmission Control Protocol (TCP)-based and User Datagram Protocol (UDP)-based IP traffic, as they possess different characteristics that have to be accounted for when dimensioning the ATM backbone. The modelling of the TCP-based traffic involves characterization of the retransmissions of TCP/IP packets as a result of transmission errors inherent to ATM networks. When large TCP/IP packets are segmented to ATM cells in order to adapt to ATM transportation, a single cell lost at a buffer or corrupted in some way, will result in retransmission of the whole packet. Thus most of the cells comprising the packet are being retransmitted even though they were error free [5]. This repeated attempt aspect of the transmission of TCP traffic produces an overhead, which has to be taken into account when dimensioning the ATM backbone. Furthermore, the modelling of both TCP-based and UDP-based traffic has to take account of the additional overhead associated with the protocols at Layer 2 and Layer 3 (i.e., the ATM Layer, and AAL5 Layer).

This paper is organized as follows: In Section 2 we introduce input parameters, notation and assumptions used for the modelling. Section 3 presents the IP over ATM model that can be used to map parameters of the input IP traffic mix to three principal parameters of the Guérin et al model [4], viz: mean, peak and burst period. Section 4 discusses approaches for incorporating the IP over ATM model into the ATM dimensioning procedures. Section 5 provides verification of the results from simulation studies. A summary and proposed directions for future research are given in Section 6.

2 Input Parameters and Notation

The IP over ATM model takes as input the traffic estimates or forecasted point-to-point traffic (i.e., IGR-to-IGR) that are categorized by: protocol type (TCP or UDP), packet sizes and packet arrival rates (packets per second). The model takes as input, the IP packet sizes measured as cells instead of bytes, in order to account for the protocol overheads applicable at Layer 2 and Layer 3. IP packets are variable in length and must be encapsulated and segmented to fit the fixed cell size requirements of ATM for transportation. The total data encapsulation overheads at the AAL5 and the ATM layer consist of: LLC+SNAP header with a length of 8 bytes, AAL5-trailer with a length of 8 bytes, PAD field with a length of (0-47) bytes and 5 byte ATM headers. The conversion of IP packet size from bytes (L_{byte}) to cells (L_{cell}) is easily obtained from $L_{cell} = \lceil \frac{L_{byte}+X}{48} \rceil$, where X denotes the protocol overhead, which is a sum of the LLC+SNAP header and the AAL5 trailer.

Notation used in the mathematical analysis is as follows:
N - IP packet length in cells
N_{MAX} - Maximum length of IP packet in cells
IP_{N_TCP} - Total number of TCP packets with length N cells per second
$P_{\mathrm{IP}}(N)$ - IP packet error probability with length N cells
p - Cell error probability
p_{bit} - Bit error probability of ATM network
$P_{\mathrm{TCP}}\{length = N\}$ - Probability that an IP packet has length N cells in the fresh TCP flow
$P'_{\mathrm{TCP}}\{length = N\}$ - Probability that an IP packet has length N cells in the total TCP flow
F_{TCP} - Fresh TCP flow mean rate (i.e., offered traffic) [cell/s]
R_{TCP} - Repeated TCP flow mean rate [cell/s]
F'_{TCP} - Total TCP flow mean rate (i.e., carried traffic) [cell/s]
F_{UDP} - Fresh UDP flow mean rate [cell/s]
F'_{UDP} - Total UDP flow mean rate [cell/s]
m'_{TCP} - Mean bit rate of the total TCP flow [bit/s]

$\sigma'^2_{m_{\mathrm{TCP}}}$ - Variance of the bit rate of the total TCP flow [bit^2/s^2]
IP_{N_UDP} - Total number of UDP packets with length N cells per second
$P_{\mathrm{UDP}}\{length = N\}$ - Probability that an IP packet has length N cells in the fresh UDP flow
$P'_{\mathrm{UDP}}\{length = N\}$ - Probability that an IP packet has length N cells in the total UDP flow
m'_{UDP} - Mean bit rate of the total UDP flow [bit/s]
$\sigma'^2_{m_{\mathrm{UDP}}}$ - Variance of the bit rate of the total UDP flow [bit^2/s^2].
m'_{IP} - Mean bit rate of the total IP flow [bit/s]
$\sigma'^2_{m_{\mathrm{IP}}}$ - Variance of the bit rate of the total IP flow [bit^2/s^2].
$R'_{peak_{\mathrm{IP}}}$ - Peak bit rate of the total IP flow [bit/s]
b'_{IP} - Burst Period of the total IP flow [cells]

Assumptions used in the mathematical analysis:

1. $P_{\mathrm{TCP}}\{length = N\}$ is known. It is derived from the input data.
2. $P_{\mathrm{UDP}}\{length = N\}$ is known. It is derived from the input data.
3. p_{bit} is known.

It follows that:
$P_{\mathrm{IP}}(N)$ can be easily derived from a Binomial distribution. An IP packet of length N cells will be repeated if at least one cell (out of N) is corrupt:

$$P_{\mathrm{IP}}(N) = \sum_{i=1}^{N} \binom{N}{i} p^i (1-p)^{N-i} = 1 - (1-p)^N$$

Cell error probability p is derived from p_{bit} from a Binomial distribution, as well. One cell will be repeated if at least one bit out of 424 bits is corrupt (1 cell = 424 bits).

3 IP over ATM Model

3.1 TCP Model for the Mean Rate

In order to characterize the effects of the transmission errors on the packet arrival rates of the TCP traffic we make use of a simple model used in ITU-T documentation to analyze repeated attempt traffic in circuit switched networks. First, we briefly review the simple model and then we extend this model, in a straightforward manner, to the case of TCP traffic being transported across an ATM network.

A Simple Model for Repeated Attempts. Consider the following simplified picture of traffic that is presented to a telecommunication network (Fig.2). On the left of the diagram, fresh calls (first attempt traffic) are presented to the network. Due to the fact that the network has finite resources, some of these

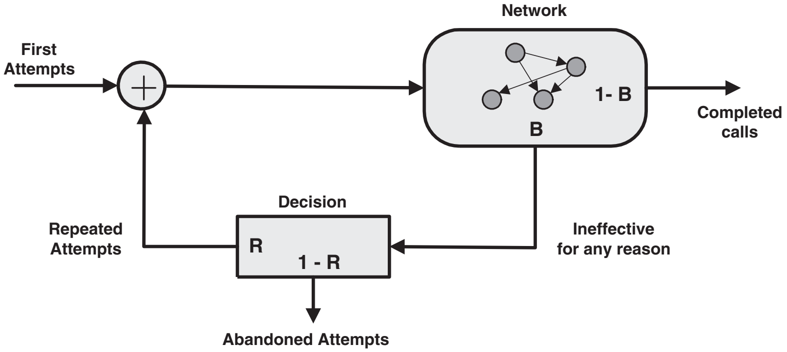


Fig. 2. Simple Model for Repeated Attempt Traffic

calls will be blocked with probability B. Conversely, the probability of call completion is given by $(1 - B)$. The ineffective calls can be abandoned or repeated. Calls are repeated with probability R and abandoned with probability $(1 - R)$. The repeated calls are added to the incoming fresh calls and represented to the network (at the '+' sign). By performing a simple analysis of this system and by defining the first attempt traffic using the symbol F and the total call attempts by T, one gets:

$$T = \frac{F}{(1 - RB)} \tag{1}$$

It should be noted that the model assumes that calls continue to repeat indefinitely until they are answered. Furthermore, the model is essentially deterministic in nature and contains no advanced stochastic modelling assumptions. Despite its simplicity, this model provides an excellent insight into the macro-operation of a circuit-switched network and the effect of repeated attempts on network performance.

Extension to TCP Traffic Carried on ATM Networks. In order to extend the model to the case of TCP traffic where packets are of variable length, we assume that a "call" is a flow consisting of TCP packets of constant length. We now give a detailed description of the new model that is based on flows that consist of TCP packets with constant length.

We define the fresh TCP flow F_{TCP} as a sum of N_{MAX} different fresh flows:

$$F_{\mathrm{TCP}} = F_{\mathrm{TCP}}(1) + F_{\mathrm{TCP}}(2) + F_{\mathrm{TCP}}(3) + \cdots + F_{\mathrm{TCP}}(N_{\mathrm{MAX}}) \tag{2}$$

where $F_{\mathrm{TCP}}(N)$ represents the fresh flow consisting of IP packets of length N cells $(N = 1, 2, 3, \ldots, N_{\mathrm{MAX}})$:

$$F_{\mathrm{TCP}} = \sum_{N=1}^{N_{\mathrm{MAX}}} F_{\mathrm{TCP}}(N) = \sum_{N=1}^{N_{\mathrm{MAX}}} N \cdot IP_{N\_\mathrm{TCP}} \tag{3}$$

As a result of the above consideration, we can apply the method of the original model to obtain a formula for $F^{'}_{\mathrm{TCP}}$. Hence, as a general case we need to analyse only the case of *one* fresh flow $F_{\mathrm{TCP}}(N)$.

The repeated flow R is simply given by the sum of N_{MAX} repeated flows caused by N_{MAX} different fresh flows: $R = R(1) + R(2) + \cdots + R(N_{\mathrm{MAX}})$, where $R(N)$ is the repeated flow caused by lost IP packets from the fresh flow $F_{\mathrm{TCP}}(N)$, $R(N) = P_{\mathrm{IP}}(N) \cdot F_{\mathrm{TCP}}(N)$. From $F^{'}_{\mathrm{TCP}} = F_{\mathrm{TCP}}(N) + R(N)$ it follows:

$$F^{'}_{\mathrm{TCP}} = \sum_{N=1}^{N_{\mathrm{MAX}}} F^{'}_{\mathrm{TCP}}(N) \tag{4}$$

Here we demonstrate a way to derive $F^{'}_{\mathrm{TCP}}$ using the original model for repeated attempts as the basis for the approach (we omit the subscript $_{\mathrm{TCP}}$ for clarity):

1-st step: $F^{'}_{1}(N) = F(N) = N \cdot IP_{N\_\mathrm{TCP}}$

$R_1(N) = P_{\mathrm{IP}}(N)F^{'}_{1}(N) = P_{\mathrm{IP}}(N)F(N)$

2-nd step: $F'_2(N) = F(N) + R_1(N) = F(N)(1 + P_{\mathrm{IP}}(N))$

$R_2(N) = P_{\mathrm{IP}}(N)F^{'}_{2}(N) = F(N)(P_{\mathrm{IP}}(N) + P^{2}_{\mathrm{IP}}(N))$

.

.

.

n-th step: $F'_n(N) = F(N) + R_{n-1}(N) = F(N)(1 + P_{\mathrm{IP}}(N) + \cdots + P^{n-1}_{\mathrm{IP}}(N))$

$R_n(N) = P_{\mathrm{IP}}(N)F^{'}_{n}(N) = F(N)(P_{\mathrm{IP}}(N) + \cdots + P^{n}_{\mathrm{IP}}(N))$

For $n \to \infty$, $F^{'}_{n}(N)$ tends to:

$$F^{'}_{\mathrm{TCP}}(N) = \frac{F_{\mathrm{TCP}}(N)}{1 - P_{\mathrm{IP}}(N)} \tag{5}$$

Using (4), for the mean cell rate of the total TCP flow one gets:

$$F^{'}_{\mathrm{TCP}} = \sum_{N=1}^{N_{\mathrm{MAX}}} \frac{F_{\mathrm{TCP}}(N)}{1 - P_{\mathrm{IP}}(N)} = \sum_{N=1}^{N_{\mathrm{MAX}}} \frac{N \cdot IP_{N\_\mathrm{TCP}}}{1 - P_{\mathrm{IP}}(N)} \tag{6}$$

Accordingly, the mean bit rate of the total TCP flow is:

$$m^{'}_{\mathrm{TCP}} = 424 \cdot F^{'}_{\mathrm{TCP}} \tag{7}$$

3.2 UDP Model for the Mean Rate

In a similar way, the fresh UDP flow F_{UDP} is modelled as a sum of N_{MAX} flows that consist of IP packets with constant length:

$$F_{\mathrm{UDP}} = \sum_{N=1}^{N_{\mathrm{MAX}}} F_{\mathrm{UDP}}(N) = \sum_{N=1}^{N_{\mathrm{MAX}}} N \cdot IP_{N\_\mathrm{UDP}} \tag{8}$$

The carried traffic consisting of IP packets of length N cells, $F^{'}_{\mathrm{UDP}}(N)$, is simply the product of the offered flow $F_{\mathrm{UDP}}(N)$ and the probability of the traffic being successfully transmitted across the network. Thus, for $F^{'}_{\mathrm{UDP}}(N)$ we have:

$$F'_{\mathrm{UDP}}(N) = N \cdot IP_{N\_\mathrm{UDP}}(1 - P_{\mathrm{IP}}(N)) \tag{9}$$

Consequently, for the mean cell rate of the total UDP flow one gets:

$$F'_{\mathrm{UDP}} = \sum_{N=1}^{N_{\mathrm{MAX}}} N \cdot IP_{N\_\mathrm{UDP}}(1 - P_{\mathrm{IP}}(N)) \tag{10}$$

The mean bit rate of the total UDP flow is simply: $m'_{\mathrm{UDP}} = 424 \cdot F'_{\mathrm{UDP}}$.

Finally, for the mean bit rate of the aggregate IP traffic carried over ATM network we have:

$$m'_{\mathrm{IP}} = m'_{\mathrm{TCP}} + m'_{\mathrm{UDP}} \tag{11}$$

3.3 Model for the Peak Rate

Since, our interest is in mapping the parameters of the aggregate IP traffic into appropriate on-off traffic parameters, we can compute the peak rate from the following on-off relation: $\sigma'^2_{m_{\mathrm{IP}}} = m'_{\mathrm{IP}}(R'_{peak_{\mathrm{IP}}} - m'_{\mathrm{IP}})$. Thus, it remains only to derive the variance of the bit rate of the aggregate IP traffic, $\sigma'^2_{m_{\mathrm{IP}}}$, which on the other hand can be expressed as a sum of the variances of the TCP and UDP flows as follows:

$$\sigma'^2_{m_{\mathrm{IP}}} = \sigma'^2_{m_{\mathrm{TCP}}} + \sigma'^2_{m_{\mathrm{UDP}}} \tag{12}$$

From the distribution of IP packet length of the fresh flow, one can derive the probabilities that an IP packet has length N cells in the total TCP flow and UDP flow respectively, in the following way:

$$P'_{\mathrm{TCP}}\{length = N\} = \frac{IP'_{N\_\mathrm{TCP}}}{\sum_{N=1}^{N_{\mathrm{MAX}}} IP'_{N\_\mathrm{TCP}}} = \frac{\frac{IP_{N\_\mathrm{TCP}}}{1-P_{\mathrm{IP}}(N)}}{\sum_{N=1}^{N_{\mathrm{MAX}}} \frac{IP_{N\_\mathrm{TCP}}}{1-P_{\mathrm{IP}}(N)}} \tag{13}$$

$$P'_{\mathrm{UDP}}\{length = N\} = \frac{IP'_{N\_\mathrm{UDP}}}{\sum_{N=1}^{N_{\mathrm{MAX}}} IP'_{N\_\mathrm{UDP}}} = \frac{IP_{N\_\mathrm{UDP}}(1 - P_{\mathrm{IP}}(N))}{\sum_{N=1}^{N_{\mathrm{MAX}}} IP_{N\_\mathrm{UDP}}(1 - P_{\mathrm{IP}}(N))} \tag{14}$$

Then, for the variance of the IP packet length for the total TCP flow $\sigma'^2_{l_{\mathrm{TCP}}}$ - we have:

$$\sigma'^2_{l_{\mathrm{TCP}}} = \sum_{N=1}^{N_{\mathrm{MAX}}} (N - \overline{N'})^2 \cdot P'_{\mathrm{TCP}}\{length = N\} \tag{15}$$

Where $\overline{N'}$ is the mean IP packet length of the total TCP flow. From (13) for the number of IP packets with length N cells in the total TCP flow we have:

$$IP'_{N\_\mathrm{TCP}} = P'_{\mathrm{TCP}}\{length = N\} \sum_{N=1}^{N_{\mathrm{MAX}}} IP'_{N\_\mathrm{TCP}} = sP'_{\mathrm{TCP}}\{length = N\} \tag{16}$$

The sum $\sum_{N=1}^{N_{\mathrm{MAX}}} IP'_{N\_\mathrm{TCP}}$ is constant, and thus we designate it using the constant s. Then, by substituting (16) in (7) for the mean bit rate one gets:

$$m'_{\mathrm{TCP}} = 424 \sum_{N=1}^{N_{\mathrm{MAX}}} N \cdot IP'_{N\_\mathrm{TCP}} = S \cdot \overline{N'} \tag{17}$$

where, for simplicity, we use $S = 424 \cdot s$. Having a relation between the mean bit rate and the mean IP packet length of the total TCP flow, one can expect that the same relation holds between the bit rate, ν_b, and the IP packet length:

$$\nu_b = S \cdot N \tag{18}$$

The variance of the bit rate for the total TCP flow - $\sigma'^2_{m_{\mathrm{TCP}}}$ can be derived from:

$$\sigma'^2_{m_{\mathrm{TCP}}} = \sum_{N=1}^{N_{\mathrm{MAX}}} (\nu_b - m'_{\mathrm{TCP}})^2 \cdot P\{\nu_b = S \cdot N\} \tag{19}$$

By applying (17) and (18) in (19) and by assuming that the probability an IP packet has length N cells is equal to the probability that the bit rate is $S \cdot N$, $P\{\nu_b = S \cdot N\}$, for the variance of the bit rate for the total TCP flow we have:

$$\sigma'^2_{m_{\mathrm{TCP}}} = S^2 \cdot \sigma'^2_{l_{\mathrm{TCP}}} \tag{20}$$

The validity of this can be shown under the assumption that a time scale is divided into time slots with constant length T and that only one IP packet enters the system in each time slot. Consequently, the corresponding flow is one packet per unit of time or $s = \frac{1}{T}$ and the corresponding bit rate is $\nu_b = 424 \cdot N \cdot s = S \cdot N$.

The variance of the bit rate of the total UDP flow can be obtained in the same way as for the TCP flow, except in the above analysis one should apply (14) instead of (13).

Finally, in order to obtain an unbiased estimator for the variance of the bit rate of the total flow - $\hat{\sigma}'^2_{m_{\mathrm{IP}}}$, (which is important because we estimate the variance directly from sample measurements) we have to multiply the estimated variance of bit rate, $\sigma'^2_{m_{\mathrm{IP}}}$, with a factor to remove the bias in this calculation. In our case, the bias factor is $\frac{C}{C-1}$ where C is the number of observations, which, in our case, is the number of total IP packets entering the system per second, $\sum_{N=1}^{N_{\mathrm{MAX}}}(IP_{N\_\mathrm{TCP}} + IP_{N\_\mathrm{UDP}})$. Since, we know the variance of the bit rate of the total flow (i.e its unbiased estimator) and the mean bit rate of the total flow (11), we can easily derive the peak bit rate of the total flow as:

$$R'_{peak_{\mathrm{IP}}} = \frac{\hat{\sigma}'^2_{m_{\mathrm{IP}}}}{m'_{\mathrm{IP}}} + m'_{\mathrm{IP}} \tag{21}$$

3.4 Model for the Burst Period

One approach to determine the burst period (e.g., maximum number of cells that can be transmitted at peak rate) is similar to the one used to dimension buffers inside IP routers. In the case of best-effort traffic, traditional rule is to dimension the buffer according to the bandwidth-delay (or *rtt*) product. In our case, where the aggregate IP traffic uses VBR service, the maximum burst size in cells is obtained as a product of the mean cell rate and the round-trip time, *rtt* :

$$b'_{\mathrm{IP}} = \frac{m'_{\mathrm{IP}}}{424} \cdot rtt \tag{22}$$

Some measurement studies have reported that such estimation works correctly for large aggregates of IP traffic [10].

In addition, to the above approach we are currently considering another approach for modelling the burst period, which is based on analysis of the User Parameter Control (UPC) functions at the ingress of ATM. Namely, we consider scenario where source is policed by two leaky buckets operating on different time-scales: one leaky bucket polices the peak rate, $R'_{peak_{\mathrm{IP}}}$, with a tolerance τ_p, and the other polices the mean rate, m'_{IP}, with a tolerance τ_m. These tolerances, τ_p and τ_m, translate as bucket sizes of the leaky bucket policers for the peak and the mean rate, respectively. Under the assumption that the source do not violate the negotiated peak rate, the maximum burst of cells that can pass through the mean rate policer is [8]:

$$b'_{IP} = \frac{\tau_m}{1 - \frac{m'_{\mathrm{IP}}}{R'_{peak_{\mathrm{IP}}}}} \tag{23}$$

In order to obtain an estimate of the tolerance τ_m, we study G/D/1/N - delay loss system, which is an exact model for the cell loss probability of the leaky bucket mechanism (if violating cells are discarded). One approximate solution for the cell loss probability of such system with on-off source model, which is sufficiently accurate in most cases, has been reported in [9]. We want to determine the cell loss probability as a function of the queue capacity (bucket size), for a given traffic load and given leaky bucket rate. The leaky rate is fixed by introducing an overdimensioning factor $C(C > 1)$, as a product of this factor and the source mean rate. The relationship between the cell loss probability and the bucket size has two distinct parts: a cell-scale component which decreases exponentially (linearly on a log scale) with the increase of the bucket size and the burst component where the slope of the curve changes dramatically as we reach certain bucket size. The intercept between these two components define solution for the required bucket size and tolerance τ_m. However, we do not present any results here as this is still an ongoing research.

4 Incorporating the Model into the ATM Dimensioning

Our model defines the aggregate IP traffic between two OD pairs (e.g. IGR-to-IGR) as a single IP traffic stream[1] and provides its characterization in terms

[1] Note that the model can be extended to incorporate QoS and thus define the aggregate IP traffic as multiple IP traffic streams. For that, it is necessary for the input traffic parameters to be categorized by QoS as well.

of the three parameters: mean rate, peak rate, and burst period. From these parameters one can calculate directly the equivalent bandwidth required for this traffic stream and thus model the aggregate IP traffic as an“IP call” with a capacity requirement equal to the equivalent bandwidth.

The equivalent bandwidth computed for the “IP call” can now be used in two different ways. The first approach is to model this aggregate IP traffic as call ariving at some average arrival rate (Poisson distributed) and persisting for an average time period (Negative exponentially distributed); it requires the nominal equivalent bandwidth for the duration of the call. This approach enables these “IP calls” to be treated in a “unified” fashion along with other ATM calls that can be characterized by their mean arival rate, service time and equivalent bandwidth. When dimensioning ATM networks using this approach, a call loss probability is specified and the capacity determined using their mean arrival and service rates in conjunction with their equivalent bandwidths. The second approach recognizes the fact that IP traffic coming from a LAN or WAN is likely to be “always on” and that we simply need to allocate capacity based on the equivalent bandwidth computed for our single “IP call” emanating from the LAN or WAN. Adopting the first approach in this case would lead to gross over-dimensioning of the capacity requirements. This means that if such traffic is present, one should carefully separate the “always on” traffic from the traffic that can be characterized by an arrival and service rate.

Finally, we consider another approach that can be taken for modeling the aggregate IP traffic stream to fit the “unified” ATM dimensioning model. We can transform our VBR “IP call” of capacity equivalent to its equivalent bandwidth into equivalent CBR “IP call” of different characteristics. According to the equivalent burst approximation [6], a general traffic stream which is the superposition of a large number of independent traffic streams can be replaced by an equivalent process with simpler characteristics. The equivalent process is chosen to have the same values for the following parameters: (1) Mean cell rate, m (2) Variance of instantaneous rate, σ^2 (3) Asymptotic variance of the number of cell arrivals in a long time interval, v. This equivalent traffic process has a Poissonian arrival process λ, for independent but equally distributed bursts with peak rate R, and bursts durations exponentially distributed with mean μ. The fitting relations are given by:

$$\lambda = \frac{2m^2}{\nu} \qquad R = \frac{\sigma^2}{m} \qquad \mu = \frac{\nu}{2\sigma^2} \tag{24}$$

Thus, by fitting the mean, the instantaneous variance, and the asymptotic variance of the aggregate IP traffic into this equivalent process, we get as a result a CBR “IP call” of capacity R, and traffic intensity $\lambda\mu$. Accordingly, it only remains the asymptotic variance of the aggregate IP traffic to be derived. The expression for the asymptotic variance of the number of offered cells in a long time interval, follows from the application of the formulae for asymptotic variance of a cumulative regenerative process [7].

5 Simulation Results

Purpose built real-time simulator was used to analyse the analytical model. The simulator is a very simple implementation of the IP over ATM model. However, it tries to test the model with more realistic data flows. The simulator tests the effects of the transmission errors on the packet arrival rates. To be able to easily understand these effects and their behaviour, the system is kept very simple by limiting it to one ATM link.

Traffic from individual IP connections is modelled as sessions which arrive according Poisson with given mean and negative exponential durations with mean values selected from a given range. The sessions are generated as TCP or UDP according to a specified probability. The packets from UDP sessions are generated deterministically, whereas the generation of packets from TCP sessions is governed by a protocol based on a TCP sliding window protocol, but has not been implemented in full detail. First a full data of window is sent (the window is set to a default value) and every other packet is sent after an arrival of ACK from the receiver. The packets are being retransmitted if an ACK has not been received within a fixed time interval. The packets from a session are of fixed length, which are selected from a given range.

The simulator gives results for the mean rate and the peak rate of the carried IP traffic on ATM link as a function of BER, as well as, for the distribution of IP packet lengths of the offered traffic, which is needed for the input of the analytical model. The results in Table 1 are obtained for input traffic of mean arrival rate equal to 40 sessions per second, out of which 90 % is TCP and 10 % UDP. The analytical model for the mean rate gives values within the range of (0.39 %, 7.25 %) from the results obtained from the simulator. The analytical model for peak rate gives results within the range of (1.43 %, 9.23 %) from the results obtained from the simulator. Discrepancies bigger than 0.55 % for the mean rate and bigger than 1.76 % for the peak rate were recorded for BER $> 10^{-6}$.

Table 1. Mean Rate and Peak Rate as a function of BER

BER	Mean (an)	Mean (sim)	an/sim	Peak (an)	Peak (sim)	an/sim
	[Mbps]	[Mbps]	%	[Mbps]	[Mbps]	%
1E-10	1.264	1.259	0.39	1.835	1.809	1.43
1E-9	1.264	1.259	0.39	1.835	1.809	1.43
1E-8	1.264	1.259	0.39	1.835	1.809	1.43
1E-7	1.265	1.260	0.40	1.836	1.810	1.46
5E-7	1.269	1.263	0.47	1.842	1.812	1.52
1E-6	1.276	1.269	0.55	1.849	1.817	1.76
5E-6	1.310	1.287	1.78	1.900	1.846	2.89
1E-5	1.359	1.310	3.74	1.972	1.893	4.15
5E-5	1.907	1.830	4.21	2.684	2.252	6.50
1E-4	3.210	2.993	7.25	4.100	3.740	9.23

6 Conclusions

In this paper we have described a mathematical model that characterises the aggregate traffic of multiple TCP/IP connections as it enters the ATM backbone into a form that is suitable for the dimensioning processes of the ATM network. Specificaly, we modelled the aggreagte IP traffic that is being carried over ATM link as an "aggregate IP call" characterized by the three principal parameters of the Guérin et al model, viz [4]: mean rate, peak rate, and mean burst period. This enabled us to calculate the effective bandwidth of the "aggregate IP call" and to use it subsequently for the dimensioning of the ATM networks.

To validate the modelling, we have developed a simulation model. The simulation results have shown that our mathematical model is very accurate in demonstrating the effects of increasing cell loss probabilities (i.e., bit error rates) on the performance of the IP traffic over ATM networks. The success of the method shows that we can translate the IP traffic measurement data into their equivalent cell-level parameters for direct application into the ATM dimensioning procedures.

Future work will involve the development of tools to implement the procedures described in this paper for optimal design of ATM networks.

References

1. Berry, L.T.M., Harris, R.J., Puah, L.K.: Methods of Trunk Dimensioning in a Multiservice Network. In Proceedings of GLOBECOM'98. (1998) 282–287
2. Kaufman, J.S.: Blocking in a Shared Resource Environment. IEEE Transactions on Communications. **29** (1981) 1474–1481
3. Roberts, J.W.: A Service System with Heterogeneous User Requirements - Application to Multi-Service Telecommunications Systems. In Proceedings of Performance of Data Communication Systems and their Applications, G. Pujolle (ed.). (1981) 423–431
4. Guérin, R., Ahmadi, H., Naghshineh, M.: Equivalent Capacity and its Application to Bandwidth Allocation in High-Speed Networks. IEEE Journal on Selected Areas In Communications. **9** (1991) 968–981
5. Hassan, M., Breen, J.: Performance Issues for TCP/IP over ATM. 7th International Network Planning Symposium - Planning Networks and Services for the Information Age. (1996) 575–580
6. Lindberger, K.: Analytical Methods for the Traffical Problems with Statistical Multiplexing in ATM Networks. In Proceedings of the 13th International Teletraffic Congress. (1991) 807–813
7. Smith, W.L.: Renewal Theory and its Ramifications. Journal of Royal Statistical Society B. **20** (1958) 243–302
8. ATM Forum. ATM Traffic Management Specification Version 4.1. (1999)
9. Butto, M., Cavallero, E., Tonietti, A.: Effectivness of the "Leaky Bucket" Policing Mechanism in ATM Networks. Journal on Selected Areas in Communications. **9** (1991) 335–342
10. Bonaventure, O.: PhD. Integration of ATM Under TCP/IP to Provide Services with Minimum Guaranteed Bandwidth. Université de Liège (1998)

Analysis and Comparison of Internet Topology Generators

Damien Magoni and Jean-Jacques Pansiot

Université Louis Pasteur – LSIIT
Pôle API, Boulevard Sébastien Brant, 67400 Illkirch, France
{magoni, pansiot}@dpt-info.u-strasbg.fr

Abstract. The modeling of Internet topology is of vital importance to network researchers. Some network protocols, and particularly multicast ones, have performances that depend heavily on the network topology. That is why the topology model used for the simulation of those protocols must be as realistic as possible. In particular a protocol designed for the Internet should be tested upon Internet-like generated topologies. In this paper we provide a comparative study of three topology generators. The first two are among the latest available topology generators and the third is a generator that we have created. All of them try to generate topologies that model the measured Internet topology. We check their efficiency by comparing the produced topologies with the topology of a recently collected Internet map.

1 Introduction

Today simulation tools are widely used to test network protocols. These tools need network topologies as input data. A network topology is usually modeled by an undirected graph where the network devices are modeled by the nodes of the graph and the communication links are modeled by the edges of the graph. A software tool that creates network topologies is usually called a graph generator or a topology generator. The way it builds network topologies (i.e. the set of creation procedures) is called a topology model.

In this paper, we will focus on some of the latest Internet-like topology generators, including one that we have created. We will assess the efficiency of these generators by comparing the graphs that they produce with a graph built from real data and representing a part of the Internet at the router level. The rest of the paper is organized as follows. Section 3 presents the Internet map that we use as a reference, and the generators that we test as well as their settings. Section 4 details the properties studied. Finally, in section 5 we give the results of the analysis of the generated graphs compared to the Internet map analysis.

2 Previous Work

The study of the Internet topology is an area of active research. There is not much information on the topology of Internet at the router level because it is very hard to obtain. An attempt to map the Internet was carried out by Pansiot *et al.* [11] by using

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 364–375, 2002.

source routing. Their data collect was done during the summer of 1995 and the resulting map contained 3888 routers. They also defined terms that we use in our work. More recently, another collect was undertaken by Govindan *et al.* using a heuristic called hop-limited probes [6]. This heuristic (and many others) has been included in their software called Mercator. Their collect was carried out in 1999 and the resulting map contained 228263 routers. Because it is easier to get exhaustive routing data, the Autonomous System (AS) level topology of Internet has been further investigated. From 1994 to 1995, a study of the Internet inter-domain topology was carried out by Govindan *et al.* [5]. In 1999, Faloutsos *et al.* [4] analyzed the inter-domain routing information provided by the NLANR and the routers' map made by Pansiot *et al.* They found that the Internet topology obey power laws at both the AS level and the router level. In [6], Govindan *et al.* noticed that some of the Faloutsos *et al.* power laws still hold for their router level Internet instance of 1999. Recently, additional power laws were found by Magoni *et al.* at the AS level [8] also by using data provided by the NLANR.

Concerning the topology generators, one of the earliest and most famous models was designed by Waxman in 1988 [12]. This kind of model is usually called flat topology model. The nodes are randomly placed on an euclidean plane irrespective of any hierarchy order among them. This model was later replaced by hierarchical topology models such as the Tiers [3] and the Transit-Stub [13] models. These models try to enforce a multi-level hierarchy that can be found in the Internet (e.g. host-router-AS). The discovery of power laws in the Internet by Faloutsos *et al.* has brought the arrival of a new kind of topology model. We call it the power law topology model because, as the name suggests, it makes use of power laws to generate Internet-like graphs. The BRITE [10] generator as well as our topology generator called network manipulator (*nem*), belong to this category and generate router level graphs. Inet2 [7] also belongs to the power law topology model category but it generates AS level graphs. Finally two models were recently defined to reflect the power laws found in huge network topologies. The first one was defined by Aiello *et al.* in [1] and the second one called Extended Scale-free model was defined by Albert *et al.* in [2].

3 Source and Tools

3.1 Internet Map

There are basically two levels in Internet topology. The router level and the AS level. Although AS level maps are easier to make, we will focus our attention on the router level maps of the Internet. The main reason for this choice is that a router level map provides a higher accuracy for IP layer protocol simulations. A simulation of a protocol designed for the IP environment should use such a map because it displays the IP connection topology.

As we said in the previous section, one of the most recent router level Internet maps was constructed by Govindan *et al.* with their software tool Mercator [6]. Their map is called **scan**. They also recovered another map built by researchers at the *Lucent* laboratories. This map is called **lucent**. Both maps were merged to create one of the most recent and complete Internet map ever built. This map is called **scan+lucent**. Govindan *et al.* have made the anonymized version of this map freely available for download.

It is this map that we have used in our study. We call it an Internet reference map. It is a huge map containing 284772 nodes and 449228 edges. When we show a property value of the **scan+lucent** map in a figure, we label it "Internet" for simplicity instead of **scan+lucent**. We are aware that the map is probably not exhaustive.

3.2 Topology Models

The three topology generators studied in this paper are BRITE, Inet2 and *nem*. All of them belong to the power law topology model (i.e. the latest and most accurate model). The graph sizes we have chosen to study are 500, 1000, 2000, 4000, 8000 and 16000 nodes. This should give a good view of the scaling effect on the properties of the generated graphs. Although researchers may need graphs smaller than 500 nodes (e.g. for resource consuming simulations), it is difficult for these generators to create very small graphs as they are based on power laws that arise only with big numbers. As Inet2 is, by design, not able to generate graphs of size below 3037, we have only created Inet2 graphs of sizes 4000, 8000 and 16000. We have generated **20** graphs of each chosen size for each topology generator.

We explain here the parameter settings that we used to generate the graphs. We have chosen to test BRITE with m = 1, 2 and 3 (it is the number of links added per new node). As we use incremental growth, we obtain graphs with an average node degree of 2m. The Internet reference map has an average node degree of 3.15, so taking m above 3 would have given graphs with too high an average node degree (i.e. it means too many edges compared with the number of nodes). Furthermore we found different results for m = 1, 2 and 3, despite what Medina *et al.* found in [10]. That is why in the result section we consider three scenarios for the use of BRITE (i.e. for m = 1, 2 and 3) as if we had three different graph generators. Given the results of the authors of BRITE, we use a random node placement because a Pareto node placement gives similar results. Also we use preferential connectivity and incremental growth both turned on because only these settings generate graphs that obey the outdegree and rank power laws as shown by Medina *et al.* in [10]. The generation method (i.e. how graphs are created) of BRITE is fully explained in [10].

Concerning Inet2, a first very important remark is that it generates **AS level** graphs. To compare it with the other generators, we simply consider Inet2 output to be router level graphs. We note that Jin *et al.* did the same in [7] when they compare Inet2 with Waxman, Tiers, Transit-Stub and BRITE that they also consider as AS level generators. We will refer back to this point when it is relevant, and sometimes compare Inet2 graphs with an AS level map of May 2000 analyzed by Magoni *et al.* in [8]. The generation method of Inet2 is fully explained in [7].

Concerning *nem*, it is worth noticing that it creates graphs by extracting a subgraph from a real Internet map. Thus we usually call it a topology modeler rather than a topology generator. We use the **scan+lucent** map as its input real Internet map. Of course it can be argued that if we compare the graphs generated by *nem* with a reference map that is its input map, the results will automatically be matching what is desired but this is not true. The process of extracting a graph of a few hundred or thousand nodes from a map having nearly 300,000 nodes can make this graph have a completely different topology than the originating map. The generation method of *nem* is fully explained in [9].

4 Properties of Interest

In this section we describe which topological properties we have chosen to study. We use the regular terminology set forth in previous papers by Pansiot *et al.* [11], Faloutsos *et al.* [4] and Magoni *et al.* [8]. From now on and for the sake of simplicity, we will talk about the property values of the *Internet* instead of the property values measured in the **scan+lucent** map (i.e. the map that we use as our reference map). (e.g. we write "the diameter of the Internet is ... " instead of "the diameter of the **scan+lucent** reference map is ... ".)

The node *outdegree* or *degree* (i.e. as we consider undirected graphs) distribution is one of the fundamental properties of a graph. The degree distribution of the Internet is a skewed distribution. From a graph's degree distribution we infer the average node degree, power law 1 (rank exponent) and power law 2 (outdegree exponent). Both power laws have been found by Faloutsos *et al.* in [4]. Medina *et al.* have shown in [10] that Waxman graphs and Transit-Stub graphs do not comply with power laws 1 and 2.

Another important property is the distance distribution. We define the distance between two nodes as being the hop count between the two (i.e. the minimum number of edges to cross to get from one node to the other). The distance is also called shortest path length (defined by a number of hops). It is worth noticing that the distance distribution of the Internet is not skewed. It seems to be Gaussian. This means that the average distance inferred from the distance distribution is a good indicator to study. The biggest distance of a given node to any other node is called the *eccentricity* of the node. The eccentricity distribution of the Internet also seems to be Gaussian and thus we study the average eccentricity.

Furthermore we will not study power law 3 (i.e. eigen exponent) found by Faloutsos *et al.* because Medina *et al.* have shown in their paper [10] that power law 3 holds for large Waxman graphs, Transit-Stub graphs and nearly all BRITE configurations. As Waxman and Transit-Stub graphs do not model the Internet topology accurately, we think that power law 3 is not a primary indicator.

From the definition given by Magoni *et al.* in [8], a node belonging to a cycle or lying on a path connecting two cycles is called an *in-mesh* node. The *mesh* is the set of all in-mesh nodes of the graph. We examine the mesh size and we give results about the mesh connectivity such as the number of cutpoints and the biggest bicomponent size. The study of the mesh gives information about the amount of reliability vs connection failures and the possibility of load balancing by using alternate paths.

Finally we examine the forest part of the graphs. We look at the number of trees and at the tree size distribution. In the Internet, we have found that power laws 6 and 7 (found by Magoni *et al.* at the AS level in [8]) can be inferred from this distribution. We examine the generated graphs to see if they also comply with these power laws. The properties concerning the trees are interesting for studying the network reliability and connectivity as each node belonging to a tree is a cutpoint (excepted the leaf nodes) and thus it can make the graph disconnected if it fails.

5 Results

This section contains the results of the analysis of all the graphs generated by the three topology generators. These results are compared with the results of the analysis that we made on the **scan+lucent** map. *In all the following figures, the value measured in the **scan+lucent** map of the given property is plotted as a dashed horizontal line. Of course it is not corresponding to the network size coordinate axis. It only serves as a reference value that can be easily compared to the values measured in the graphs.* Any property value for a generator for a given size is the average of the values measured for each of the 20 generated graphs. For instance, the average node degree of the 500-node graphs generated by BRITE with `m` = 2 is 3.6. This means that this value is the average of the average node degree of each of the 20 graphs of size 500 generated with BRITE (`m` = 2). In what follows we will write BRITE x instead of BRITE with `m` = x.

5.1 Degree Properties

Figure 1 shows the plots of the average node degree. The first striking observation is that it is just below 2 for the BRITE 1 graphs. In fact we checked that each graph of size n generated with BRITE 1 has exactly $n - 1$ edges and is connected. This means that these graphs are merely trees. This is confirmed by the authors of BRITE who state that for a newly considered node they connect it with only one link [10] when `m` = 1. BRITE 3 graphs have an average degree between 5.5 and 6, which is nearly the double of the Internet average degree value. However, as the degree distribution is skewed, this may not be of great importance. Finally, *nem* has exactly the same average degree as the Internet. This is because, as we saw in section 3, it generates the amount of edges needed to match the Internet average degree. It is worth noticing that the average node degree of the graphs from all the generators does not depend much on the size of the graph (excepted for 2000-node or less BRITE 3 graphs).

Figure 2 shows the plots of the absolute correlation coefficient (ACC) of the average degree distribution with respect to power law 2 (i.e. outdegree exponent). We clearly see that BRITE 2 and BRITE 3 graphs do not comply with power law 2. We also see that Inet2 graph ACCs tend to decrease when the size increases. In particular the graphs of size 16000 have an average ACC a little under 0.95. Finally, *nem* and BRITE 1 graphs have very good ACCs that decrease a little when the graph size is small (e.g. 500).

We have examined some samples of the degree distributions of BRITE 2 and 3 graphs. Here is an example of the beginning of a degree distribution of a 4000-node BRITE 3 graph:

Degree	Frequency	Frequency in %
1	5	0.125
2	4	0.1
3	1601	40.025
4	766	19.15
5	450	11.25
..	..	..

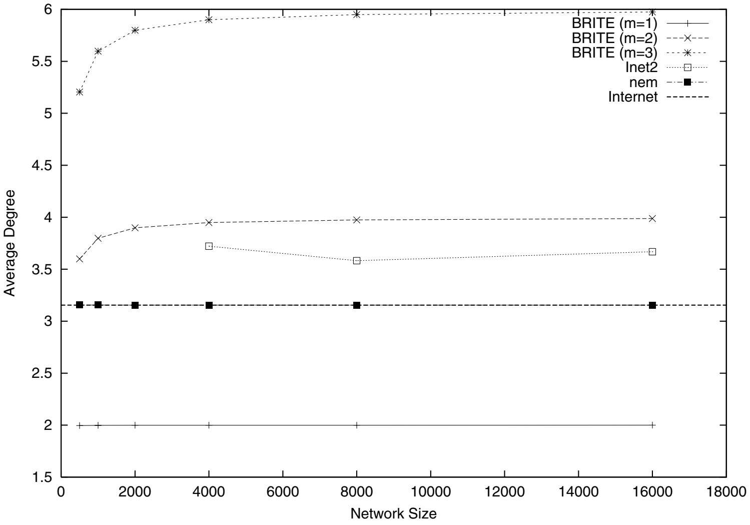


Fig. 1. Average Degree

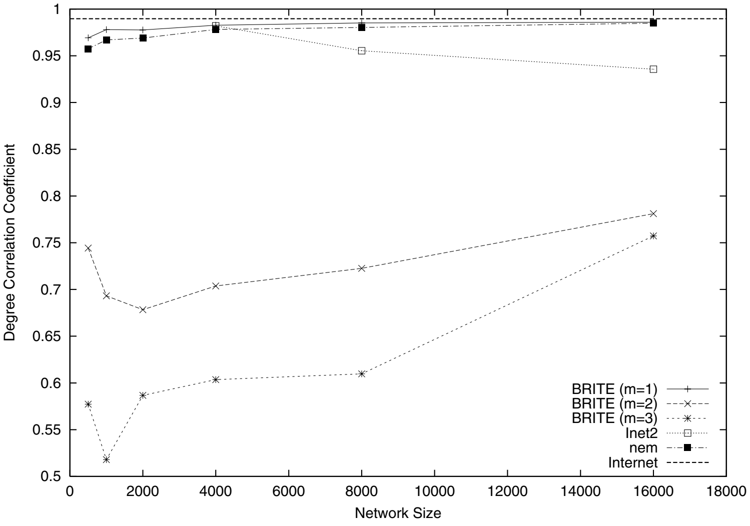


Fig. 2. Degree Correlation Coefficient

It is the few degree 1 and degree 2 nodes that cause these graphs to *not* comply with power law 2. We verified for BRITE 2 and 3 graphs that the BRITE algorithm generates a degree distribution that starts at `m` instead of one (when `m` is greater than 1) and a few outliers of degree less than `m` cause BRITE graphs to *not* follow power law 2. We do not show the plots of the absolute correlation coefficient (ACC) of the rank distribution because all generators comply with power law 1 (i.e. rank exponent).

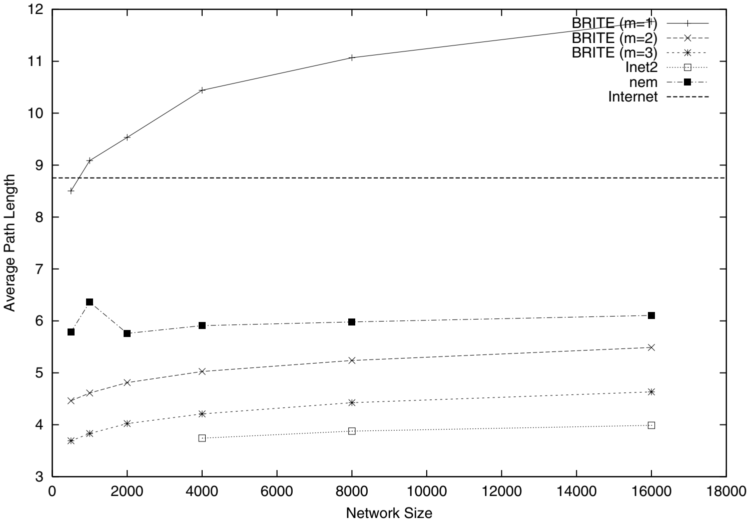


Fig. 3. Average Path Length

To conclude the study on the degree distribution of the graphs, we can already see that many factors contribute to the realism of a graph generator. We saw here that BRITE 1 complies with power laws 1 and 2, but its graphs are trees and thus do not match the Internet topology at all (particularly in terms of redundant links). BRITE 2 and 3 do not comply with power law 2 because of the way they shift the degree distribution (i.e. the majority of the nodes should have degree 1 and not degree `m`). Inet2 and *nem* generate graphs that bear a closer similarity to the Internet degree properties.

5.2 Distance Properties

Figure 3 shows the plots of the average distance (i.e. path length). Except for BRITE 1 graphs, all graphs have an average distance below the Internet average distance which is 8.75. BRITE 1 average distance increases when the graph size increases presumably because of its tree structure. The average distance of the others does not seem to vary with changes in size. Inet2 graphs have the lowest values (all below 4) but this is surely due to Inet2 design (i.e. an AS level generator). Magoni *et al.* found that the average distance of the AS level Internet in May 2000 was 3.65 which is very close to the average distance values of Inet2 graphs. As distances are an important factor in simulations (particularly for determining delays) it would be desirable to have average distances as close as possible to the Internet average distance of 8.75. *nem* graphs with average distance values around 6 and BRITE 1 graphs with values around 11 (although much influenced by the graph size) are the most realistic ones.

Figure 4 shows the plots of the average node eccentricity. Here BRITE 2 and 3 graphs have values not influenced by the graph size and below 8, which is far from the Internet eccentricity value of nearly 20. BRITE 1, Inet2 and *nem* values seem to depend

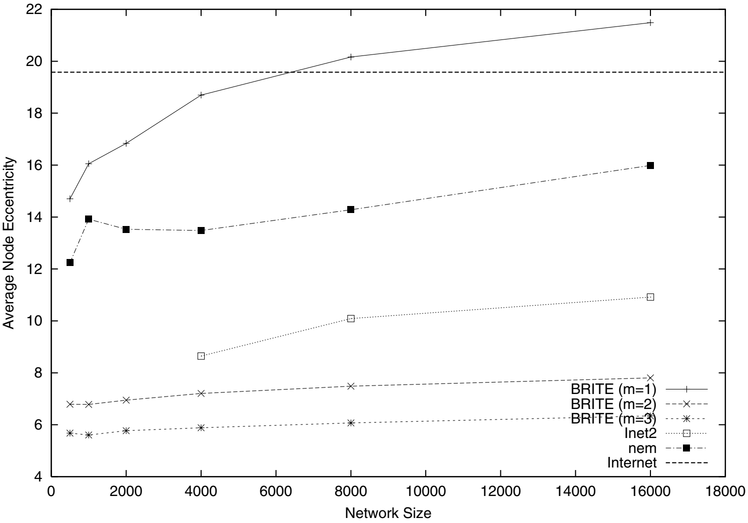


Fig. 4. Average Node Eccentricity

on the graph size. When the graph size increases, the eccentricity increases. However only BRITE 1 seems to match the Internet eccentricity. Inet2 has values that are too low (i.e. below 11) and *nem* values although reaching 16 for 16000-node graphs are still 25% under the Internet eccentricity value. Eccentricity gives an idea of the spreading of the graph. If it is low it means that there are only a few nodes that are far removed from the others, most of them being concentrated in a small area. On the other hand, a high eccentricity (as in the Internet) means that the nodes are spread over a wide area. It is a measure of how far a node is from all the other nodes. In the Internet a node is, on average, at most 20 hops from all the others. At the AS level, the eccentricity of the Internet is 7 which is well under the Inet2 graph values (especially for 16000-node graphs with a fraction of 0.35, equal to the AS level fraction, where the eccentricity reaches 11).

To conclude, we can say that distance properties are hard to model with accuracy. Average values such as distance and eccentricity are usually too low in the generated graphs compared to the Internet values. The increase in the values of BRITE 1, Inet2 and *nem* when the graph size increases is a good sign that taking bigger graphs will give adequate values. However we think that it is important to try to do better because these two properties are based on Gaussian distributions and thus are good indicators of graph realism. In fact we are currently creating a filter in our *nem* generator that allows us to generate graphs with average distances nearly equal to those measured in the Internet (i.e. between 9 and 11). It works when generating graphs having 500 nodes or more. This filter will be described in a future work.

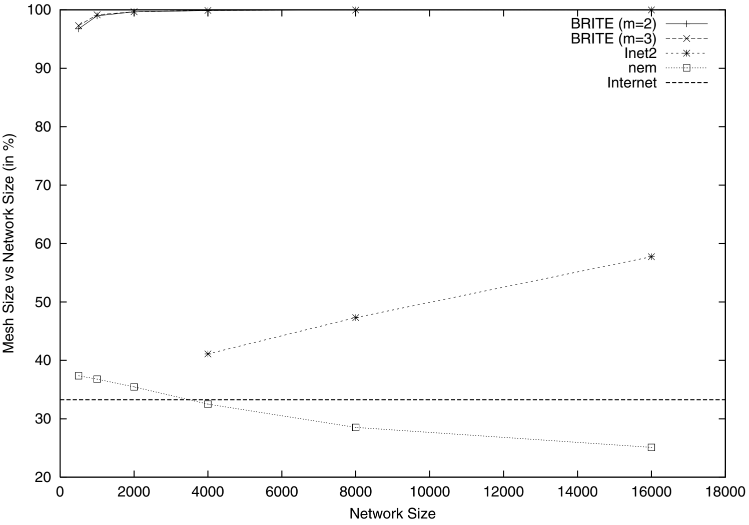


Fig. 5. Mesh Size vs Network Size (in %)

5.3 Mesh and Trees Properties

We have given the definition of the mesh in section 4. As BRITE 1 graphs are trees, they do not have a mesh and thus they will not be studied in this section. Figure 5 shows the plots of the mesh size in percentage of the graph size. The Internet has a mesh whose size represents 33% of the size of the whole Internet graph. This means that one node out of three in the Internet belongs to the mesh. We can already make a striking observation. BRITE 2 and 3 graphs have an average mesh proportion of nearly 100% ! (the lowest value is 96.74% for 500-node BRITE 2 graphs). This does not coincide with the Internet mesh proportion of 33%. Inet2 has values starting at 41% and increasing accordingly with the graph size. *nem* on the other hand has values starting at 37% and decreasing when the graph size increases. Nevertheless *nem* graphs seem to have the closest mesh proportion values to the Internet mesh proportion. It is worth noticing that Inet2 mesh proportions are accurate (particularly for 16000-node graphs) when compared with the AS level mesh proportion which equals 63%.

We have also studied the connectivity properties of the graph meshes and the Internet mesh. The number of cutpoints in the mesh of the Internet is equal to 3.7% of the total number of nodes in the mesh. In all the graphs that we studied this ratio was at most 0.39% which is much less than the Internet value. Cutpoints are important because the failure of a cutpoint router leads the network to be disconnected. Although we saw that cutpoints are rare, it is interesting to see how they partition the mesh. To examine this point, we calculated the sizes of the biggest biconnected component of the meshes. In the Internet, the biggest bicomponent contains 87% of the nodes of the mesh. This means that although the Internet mesh contains 3.7% cutpoints, these nodes only fraction a small part of the mesh (i.e. 13%), as the larger part is biconnected. Concerning the graphs, we saw that except for 500-node *nem* graphs whose average biggest bicomponent size is 95.8% of the mesh, all the other graphs have a ratio above 97.8%. To conclude we can say that the generated graphs have almost entirely biconnected meshes, which is not really the case of the Internet mesh that still contains a few cutpoint nodes and bridge nodes (i.e. nodes not belonging to a cycle but on a path linking two cycles).

The forest is simply the set of nodes of the graph that do not belong to the mesh. These nodes are located in trees and the union of these trees form the forest. The trees are connected to the mesh by special nodes called roots. We consider the roots as nodes belonging to the mesh. As we said before, BRITE 1 graphs are trees. Any one BRITE 1 graph of size n has exactly $n - 1$ edges and is connected hence it is a unique tree. This implies that BRITE 1 graphs are always composed of one tree and thus the assumption that the graphs have forests with multiple trees is false for BRITE 1 graphs. Hence we will not study BRITE 1 graphs in the rest of this section.

Figure 6 shows the plots of the number of trees (i.e. the number of roots) *vs* the number of nodes in the graph. This is an interesting measure as root nodes represent the connection points to the in-tree nodes of the graph. We can see that BRITE 2 and 3 graphs have nearly 0% trees except for small sized ones. This is a consequence of what we saw in the previous section: nearly all the nodes of BRITE 2 and 3 graphs belong to the mesh part. When we look closer at these graphs we see that they only have a handful of trees, most of them having depth one (i.e. nodes are directly connected to the roots). Hence we have not been able to calculate tree power laws for BRITE 2 and 3 graphs and thus we do not study them in the rest of this section. In the Internet the ratio of the trees *vs* the Internet size is 11.7% (more than one in-mesh node out of three is a root node). We can see that Inet2 and *nem* graphs have roughly similar values. For small sizes, *nem* graphs have a higher percentage because their mesh is smaller and thus we find more trees. It is worth noticing that Inet2 values, already above *nem* values, are far from the AS level tree percentage which equals 7.7%.

In the following of this section we examine the presence of tree power laws [8] defined by Magoni *et al.* in Inet2 and *nem* graphs. We found that the Internet complies with power laws 6 and 7 (i.e. tree rank exponent and tree size exponent) with a high degree of accuracy. It is worth noticing that Magoni *et al.* have already found that the AS level of the Internet complies with these tree power laws.

We do not show the plots of the tree size ACCs of *nem* and Inet2 graphs because we found that the values are close to or above 0.95. Thus all these graphs comply with power law 7 (tree size exponent).We also do not show here the plots of the tree rank ACCs of *nem* and Inet2 graphs because they are all above 0.97. Thus they all closely comply with power law 6 (tree rank exponent).

To summarize this section, we can notice that only Inet2 and *nem* graphs comply with the power laws concerning the trees. Their proportion of trees matches the Internet tree proportion. Because they do not have a sizeable number of trees, the tree size distributions of the BRITE 2 and 3 graphs cannot be used to calculate the ACCs and thus they do not comply with tree power laws 6 and 7. As both the AS level and the router level of the Internet comply with the tree power laws, we think that these laws are valid indicators of a graph topology similarity to the Internet topology.

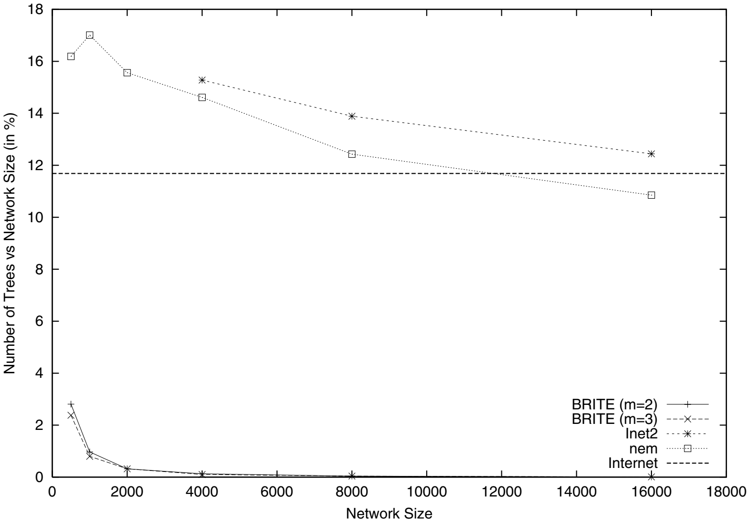


Fig. 6. Number of Trees *vs* Network Size (in %)

6 Conclusions

Designing Internet-like topology generators is not an easy task. The generated graphs have to comply with many laws and their topological properties must have suitable values that match the Internet ones. Ensuring that important power laws and Gaussian distribution averages are accurately reproduced is already a big step towards making a graph topology similar to the Internet one. In particular we think that the tree proportion and the tree power laws are interesting indicators for assessing the reliability of an Internet-like generated graph. The aims of our paper can be summarized as follows:

- Compare some of the latest generators that belong to the power law topology model, as a means to evaluating their performance.
- Analyze the generated graphs to examine how well their properties compare with the ones measured in the **scan+lucent** router level Internet map, as a means to evaluating their accuracy.
- Give, at the same time, results on the Internet topology properties inferred from the **scan+lucent** map (one of the biggest available Internet maps).

Our topology generator (*nem*) holds positive results and we hope that it will be helpful to the research community. The complexity of the Internet topology opens up the prospect of finding new properties and creating new generators in the near future.

References

1. William Aiello, Fan Chung, and Linyuan Lu. A random graph model for massive graphs. In *Proceedings of ACM STOC'00*, pages 171–180, 2000.

2. Réka Albert and Albert-László Barabási. Topology of evolving networks: local events and universality. *Physical Review Letters*, (85):5234, 2000.
3. Matthew Doar. A better model for generating test networks. In *Proceedings of IEEE GLOBECOM'96*, November 1996.
4. Michalis Faloutsos, Petros Faloutsos, and Christos Faloutsos. On power-law relationships of the internet topology. In *Proceedings of ACM SIGCOMM'99*, Cambridge, Massachusetts, USA, September 1999.
5. Ramesh Govindan and Anoop Reddy. An analysis of internet inter-domain topology and route stability. In *Proceedings of IEEE INFOCOM'97*, Kobe, Japan, April 1997.
6. Ramesh Govindan and Hongsuda Tangmunarunkit. Heuristics for internet map discovery. In *Proceedings of IEEE INFOCOM'00*, Tel Aviv, Israël, March 2000.
7. Cheng Jin, Qian Chen, and Sugih Jamin. Inet: Internet topology generator. Technical Report CSE-TR-433-00, University of Michigan, 2000.
8. Damien Magoni and Jean-Jacques Pansiot. Analysis of the autonomous system network topology. *Computer Communication Review*, 31(3):26–37, July 2001.
9. Damien Magoni and Jean-Jacques Pansiot. Internet topology analysis and modeling. In *Proceedings of IEEE Computer Communications Workshop*, Charlottesville, Virginia, U.S.A., October 2001.
10. Alberto Medina, Ibrahim Matta, and John Byers. On the origin of power laws in internet topologies. *ACM Computer Communication Review*, 30(2), April 2000.
11. Jean-Jacques Pansiot and Dominique Grad. On routes and multicast trees in the internet. *ACM Computer Communication Review*, 28(1):41–50, January 1998.
12. Bernard Waxman. Routing of multipoint connections. *IEEE Journal on Selected Areas in Communications*, 6(9):1617–1622, December 1988.
13. Ellen Zegura, Kenneth Calvert, and Michael Donahoo. A quantitative comparison of graph-based models for internetworks. *IEEE / ACM Transactions on Networking*, 5(6):770–783, December 1997.

Energy Efficient Design of Wireless Ad Hoc Networks

Carla-Fabiana Chiasserini[1], Imrich Chlamtac[2], Paolo Monti[1], and Antonio Nucci[1]

[1] Dipartimento di Elettronica, Politecnico di Torino, Corso Duca degli Abruzzi 24,
10129 Torino, Italy
{chiasserini,nucci}@polito.it

[2] Erik Jonsson School of Engineering and Computer Science,
University of Texas at Dallas, Dallas, USA
chlamtac@utdallas.edu

Abstract. One of the most critical issues in wireless ad hoc networks is represented by the limited availability of energy within network nodes. The time period from the instant when the network starts functioning to the instant when the first network node runs out of energy, the so-called network *life-time*, strictly depends on the system energy efficiency. Our objective is to devise techniques to maximize the network life-time in the case of cluster-based systems, which represent a significant sub-set of ad hoc networks. We propose an original approach to maximize the network life-time by determining the optimal clusters size and the optimal assignment of nodes to cluster-heads. The presented solution greatly outperforms the standard assignment of nodes to cluster-heads, based on the minimum distance criterion.

1 Introduction

One of the major challenges in the design of ad hoc networks is that energy resources are significantly more limited than in wired networks. Recharging or replacing the nodes battery may be inconvenient, or even impossible in disadvantaged working environments. This implies that the time during which all nodes in the ad hoc network are able to transmit, receive and process information is limited; thus, the network *life-time* becomes one of the most critical performance metrics [1,2].

Here, we define the network life-time as the time spanning from the instant when the network starts functioning to the instant when the first network node runs out of energy. In order to maximize the life-time, the network must be designed to be extremely energy-efficient. Various are the possible network configurations, depending on the application. In this paper, we deal with system architectures based on a clustering approach [3,4,5], which represent a significant sub-set of ad hoc networks.

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 376–386, 2002.

In cluster-based systems, network nodes are partitioned into several groups. In each group, one node is elected to be the cluster-head, and act as local controller, while the rest of the nodes become ordinary nodes (hereinafter nodes). The cluster size is controlled by varying the cluster-head's transmission power. The cluster-head coordinates transmissions within the cluster, handles intercluster traffic and delivers all packets destined to the cluster; it may also exchange data with nodes that act as gateways to the wired network.

In cluster-based network architectures, the life-time is strongly related to cluster-heads' failure. Indeed, power consumption in radio devices is mainly due to the following components: digital circuitry, radio transceiver, and transmission amplifier. Thus, energy consumption increases with the number of transmitted/received/processed packets and with the device's transmission range. Consider a network scenario where all nodes within a cluster are one-hop away from the cluster-head, as it often occurs in cluster-based systems [5,6,7], and assume that the traffic load is uniformly distributed among the nodes. Since cluster-heads have to handle all traffic generated by and destined to the cluster, they have to transmit, receive and process a significant amount of packets (much larger than for ordinary nodes), which depends on the number of controlled nodes. In addition, while transmitting the collected traffic to other cluster-heads or to gateway nodes, they have to cover distances that are usually much greater than the nodes' transmission range. Cluster-heads therefore experience high energy consumption and exhaust their energy resources more quickly than ordinary nodes do. The life-time of cluster-based networks thus becomes the time period from the instant when the network starts functioning to the instant at which the first cluster-head runs out of energy. In order to maximize the system life-time, it is imperative to find network design solutions that optimize the cluster-heads' energy consumption.

The procedure of cluster formation consists of two phases: cluster-head election and assignment of nodes to cluster-heads. Although several algorithms have been proposed in the literature, which address the problem of cluster formation [2,3,5,6,7,8,9], little work has been done on energy-efficient design of cluster-based networks. In [2], an energy-efficient architecture for sensor networks has been proposed, which involves a randomized rotation of the cluster-heads among all the sensors and an assignment of nodes to clusters based on the minimum distance criterion. Cluster-heads rotation implies that the network energy resources are more evenly drained and may result in an increased network life-time. On the other hand, cluster-heads re-election may require excessive processing and communications overhead, which outweigh its benefit. Thus, having fixed the nodes that act as cluster-heads, it is important to optimize the assignment of nodes to cluster-heads in such a way that cluster-heads' energy efficiency is maximized.

In this paper, we consider a network scenario where cluster-heads are chosen a priori and the network topology is either static, like in sensor networks, or slowly changing. We propose an original solution, called *ANDA (Ad hoc Network Design Algorithm)*, which maximizes the network life-time while providing the total coverage of the nodes in the network. ANDA is based on the concept that cluster-heads can dynamically adjust the size of the clusters through power control, and, hence, the number of controlled nodes per cluster. ANDA takes into account power consumption due to both the transmission amplifier and the transmitting/receiving/processing of data packets, and it levels the energy consumption over the whole network. Energy is evenly drained from the cluster-heads by optimally balancing the cluster traffic loads and regulating the cluster-heads' transmission ranges.

2 The Network Life-Time

We consider a generic ad hoc network architecture based on a clustering approach. The network topology is assumed to be either static, like in sensor networks, or slowly changing. Let $S_C = \{1, \ldots, C\}$ be the set of cluster-heads and $S_N = \{1, \ldots, N\}$ be the set of ordinary nodes to be assigned to the clusters. Cluster-heads are chosen a priori and are fixed throughout the network life-time, while the coverage area of the clusters is determined by the level of transmission power used by the cluster-heads.

Three are the major contributions to power consumption in radio devices: *i)* power consumed by the digital part of the circuitry; *ii)* power consumption of the transceiver in transmitting and receiving mode; *iii)* output transmission power. Clearly, the output transmission power depends on the devices' transmission range and the total power consumption depends on the number of transmitted and received packets. Under the assumption that the traffic load is uniformly distributed among the network nodes, the time interval that spans from the time instant when the network begins to function until the generic cluster-head i runs out of energy, can be written as

$$L_i = \frac{E_i}{\alpha r_i^2 + \beta |n_i|} , \tag{1}$$

where E_i is the initial amount of energy available at cluster-head i, r_i is the coverage radius of cluster-head i, n_i is the number of nodes under the control of cluster-head i, and α and β are constant weighting factors. In (1), the two terms at the denominator represent the dependency of power consumption on the transmission range and on the cluster-head transmitting/receiving activity, respectively. Notice that, for the sake of simplicity, the relation between the cluster-head power consumption and the number of controlled nodes is assumed to be linear; however, any other type of relation could have been considered as well, with minor complexity increase.

Considering that the limiting factor to the network life-time is represented by the cluster-heads' functioning time, the lifetime can be defined as [1,2]

$$L_S = \min_{i \in S_C} \{L_i\} \ . \tag{2}$$

Our objective is to maximize L_S while guaranteeing the coverage of all nodes in the network.

3 Energy-Efficient Network Design

In this section, we formally describe the problem of maximizing the network life-time. Two different working scenarios are analyzed: static and dynamic. In the former, the assignment of the nodes to the cluster-heads is made only once and maintained along the all duration of the system. In the latter, the network configuration can be periodically updated in order to provide a longer network life-time. Then, we propose an energy-efficient design algorithm, so-called *ANDA (Ad hoc Network Design Algorithm)*, which maximizes the network life-time by fixing the optimal radius of each cluster and the optimal assignment of the nodes to the clusters. ANDA is optimum in the case of the static scenario and can be extended to the dynamic scenario by using a heuristic rule to determine whether at a given checking time the network needs to be reconfigured.

3.1 Problem Formalization

We assume that the following system parameters are known: number of cluster-heads (C), number of nodes in the network (N), location of all cluster-heads and nodes, and initial value of the energy available at each cluster-head[1].

Let d_{ik} be the Euclidean distance between cluster-head i and node k ($i = 1, \ldots, C$; $k = 1, \ldots, N$); we have that $r_i = d_{ij}$ when j is the farthest node controlled by cluster-head i. Next, let us introduce matrix $\boldsymbol{L}$=$\{l_{ij}\}$, whose dimension is equal to $|S_C| \times |S_N|$ and where each entry l_{ij} represents the life-time of cluster-head i when its radius is set to $r_i = d_{ij}$ and it covers $n_{ij} = \{\, k \in S_N \mid d_{ik} \leq d_{ij}\}$ nodes. We have

$$l_{ij} = \frac{E_i}{\alpha d_{ij}^2 + \beta |n_{ij}|} \ . \tag{3}$$

Once matrix $\boldsymbol{L}$ is computed, the optimal assignment of nodes to cluster-heads is described by the binary variable x_{ij}. x_{ij} is equal to 1 if cluster-head i covers node j and equal to 0 otherwise. We derive the value of x_{ij} ($i = 1, \ldots, C$; $j = 1, \ldots, N$) by solving the following *max/min* problem

$$\begin{array}{rrl} \textit{maximize} & L_S & \\ \textit{subject to} & \sum_i x_{ij} \geq 1 & \forall j \in S_N \\ & L_S \leq l_{ij} x_{ij} + M(1 - x_{ij}) & \forall i \in S_C,\, j \in S_N \\ & x_{ij} \in \{0, 1\}, L_S \geq 0 & \forall i \in S_C,\, j \in S_N \ . \end{array} \tag{4}$$

The first constraint in the problem requires that each node is covered by one cluster-head at least; the second constraint says that if node j is assigned to cluster-head i, the system can not hope to live more than l_{ij}. When node j is not assigned to cluster-head i, this constraint is relaxed by taking a sufficiently large M.

[1] Notice that in the case of static nodes, this information needs to be collected only once when the network starts functioning; therefore, we neglect the cost of such an operation.

This model can be easily extended to the dynamic scenario by dividing the time scale into time steps corresponding to the time instants at which the network configuration is recomputed. Time steps are assumed to have unit duration. Then, we replace x_{ij} with x_{ij}^s, where x_{ij}^s is equal to 1 if and only if cluster-head i covers node j at time step s and 0 otherwise, and E_i, d_{ij}, n_{ij}, l_{ij} with E_i^s, d_{ij}^s, n_{ij}^s, l_{ij}^s, i.e., with the corresponding values computed at time step s. Note, however, that in this case the model is no longer linear, since the model parameters depend on the time step and, thus, on the former nodes assignment.

```
begin Covering
 for(every $j \in S_N$)
  set $max = 0$
  for(every $i \in S_C$)
   if($l_{ij} \geq max$)
    set $max = l_{ij}$
    set $sel = i$
   end if
   Cover node $j$ with cluster-head $sel$
  end for
 end for
end Covering

begin Reconfigure
 for(every $i \in S_C$)
  set $E_i$ = initial energy of cluster-head $i$
  for(every $j \in S_N$)
   Compute $d_{ij}$, $|n_{ij}|$, $l_{ij}$
  end for
 end for
 $L_S^{(new)} = L_S^{(old)} = L_S$
 $\Delta = 0$
 while($L_S^{(new)} <= L_S^{(old)} - \Delta$)
  $\Delta = \Delta + 1$
  for(every $i \in S_C$)
   for(every $j \in S_N$)
    Recompute $E_i = E_i - \Delta(\alpha r_i^2 + \beta|n_{ij}|)$
    Update $l_{ij}\ \forall i \in S_C, j \in S_N$
   end for
  end for
  Call Covering and update $L_S$
  $L_S^{(new)} = L_S$
 end while
end Reconfigure
```

Fig. 1. Pseudo-code of the network design algorithm.

3.2 ANDA: The Ad Hoc Network Design Algorithm

In order to solve the *max/min* problem described in the previous section, we introduce an algorithm, named *ANDA*, based on a novel node assignment strategy. ANDA solves to optimality the *max/min* problem in the case of the static scenario and guarantees good performance in the case of the dynamic scenario. The algorithm is composed of two main functions: the *Covering* and the *Reconfigure* procedures, where *Reconfigure* is used in the dynamic scenario only. The pseudo-code of the two functions is reported in Fig. 1.

The procedure *Covering* performs the assignment of nodes to cluster-heads by associating each node to the cluster-head that presents the longest functioning time. Thus, node j ($j = 1, \ldots, N$) will be covered by cluster-head i if $l_{ij} = \max_{k \in S_C}\{l_{kj}\}$. The resulting network configuration guarantees that energy consumption is minimized; optimality of the *Covering* procedure can be easily proved from the following consideration. Suppose that in an optimal network configuration, node j is covered by cluster-head i and that $l_{ij} < l_{hj}$ with $l_{hj} = \max_k\{l_{kj}\}$. By assigning node j to cluster-head i instead of assigning the node to h, we would obtain a shorter life-time and therefore the configuration would not be optimal.

In the dynamic scenario, the rule adopted to determine the time instants at which the network needs to be reconfigured is of crucial importance. We assume that at the time of network deployment all cluster-heads are equipped with the same amount of energy. The initial node assignment is obtained from the *Covering* procedure, which gives the optimal network configuration. However, while the system is running, each cluster-head experiences a different energy consumption depending on the number of controlled nodes and on the coverage area. By scheduling periodical node re-assignments based on the recomputed values of E_i ($i = 1, \ldots, C$), we can level the system energy consumption. Through function *Reconfigure*, we compute the new value of the available energy at cluster-head i ($i = 1, \ldots, C$) as

$$E_i^{(new)} = E_i{}^{(old)} - \Delta(\alpha r_i^2 + \beta|n_i|) , \tag{5}$$

where Δ is the time interval elapsed from the last update of the network configuration. By using $E_i^{(new)}$ and recomputing matrix $\boldsymbol{L}$, from the procedure *Configure* we obtain a new nodes assignment and a new maximized value for L_S. If the difference between the old value and the new value of L_S is greater than Δ, it is worthwhile updating the network configuration and therefore the nodes re-assignment is performed.

We point out that in ANDA the assignment of nodes to cluster-heads is obtained by determining for every node i ($i = 1, \ldots, N$) the maximum value among entries l_{ij} ($j = 1, \ldots, C$). Therefore, the complexity of the assignment procedure is $O(C \cdot N)$.

4 Numerical Results

The performance of ANDA is derived in terms of network life-time and variance of the residual energy at the cluster-heads measured at the time instant at which

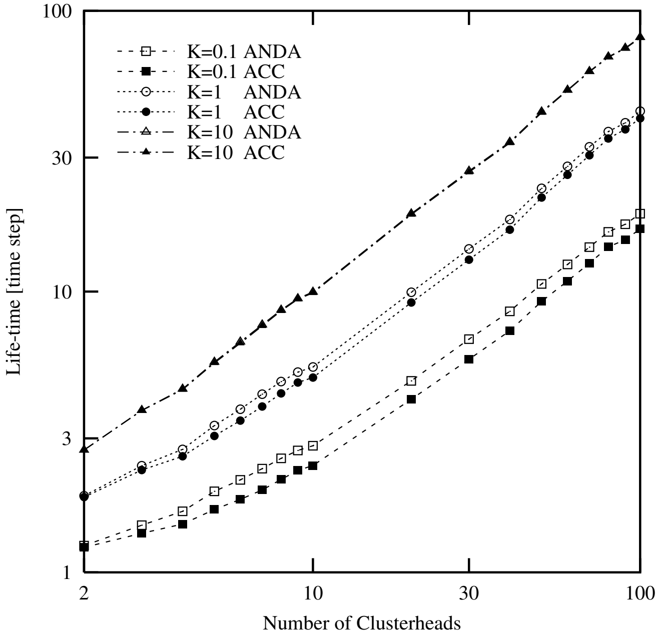


Fig. 2. Static scenario: Life-time as a function of the number of cluster-heads, for a number of nodes equal to 1000 and different values of K. Results obtained through ANDA and the ACC scheme are compared.

the first cluster-head runs out of energy. Results are plotted as functions of the ratio of the output transmission power to the power consumption due to the transmitting/receiving activity, denoted by K. We consider that all the nodes in the network are fixed and have initial energy $E_i = 1$ with $i = 1, ..., N$. We assume that the cluster-heads are uniformly distributed over the network area and are known a priori. Results were derived also in the case of a slowly changing network topology; however, they do not significantly differ from those obtained in the case of a network with fixed nodes.

First, we consider the static scenario, where only one network configuration is allowed. We compare the performance of ANDA with the results obtained by using a simple network design algorithm based on the minimum distance criterion (in the plots denoted by label *ACC (Assignment to Closest Cluster-head)*), which simply assigns each node to the nearest cluster-head. Fig. 2 shows the network life-time as a function of the number of cluster-heads, C. Curves are obtained for $N = 1000$, varying values of K, and nodes uniformly distributed over the network area. As expected, the life-time increases with the increase of the number of cluster-heads. From the comparison with the performance of the ACC scheme, we observe that the improvement achieved through ANDA is equal to 15% for $K = 0.1$, while it becomes negligible for $K = 10$, i.e., when the output transmission power contribution dominates. For both the ACC scheme

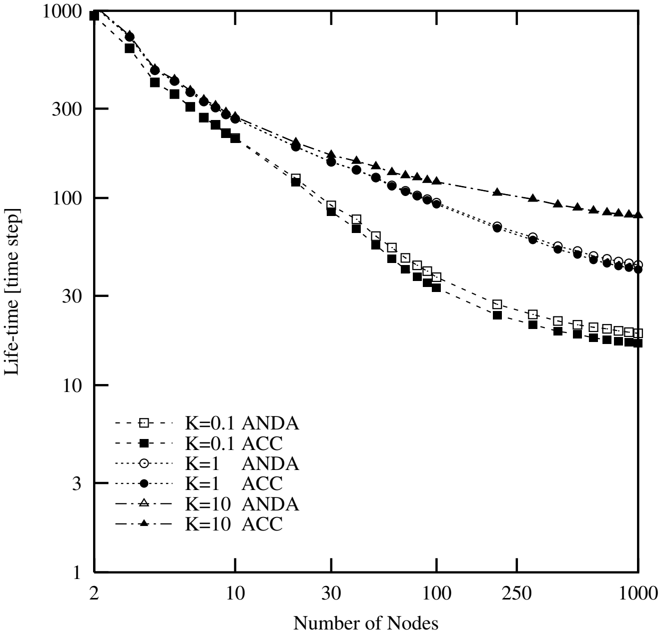


Fig. 3. Static scenario: Life-time as a function of the number of nodes, for a number of cluster-heads equal to 100 and different values of K. Results obtained through ANDA and the ACC scheme are compared.

and ANDA, a longer life-time is obtained when the major contribution to power consumption is due to the output transmission power ($K = 10$). In fact, both the schemes are able to level the output transmission power consumption among the cluster-heads; while, it is difficult to achieve an even distribution of the nodes among the clusters.

Fig. 3 shows the network life-time as the number of nodes changes, for a number of cluster-heads $C = 100$ and a uniform distribution of the network nodes. The life-time decreases as the number of nodes grows; however, for a number of nodes greater than 100, the life-time remains almost constant as the number of nodes increases.

Fig. 4 shows the variance of the residual energy at the cluster-heads as a function of the number of cluster-heads. The number of nodes in the network is set equal to 1000. For small values of C, we have a low variance since all cluster-heads have to control a large number of nodes. Increasing C, some cluster-heads may have to cover few nodes while others may experience a significant energy consumption, thus resulting in higher values of variance. For values of C greater than 25, the variance drops below 0.07 suggesting that all cluster-heads are evenly drained. Also, we notice that for small values of C and $K < 1$ we have lower variance than for $K \geq 1$ since, as mentioned above, it is hard to achieve an

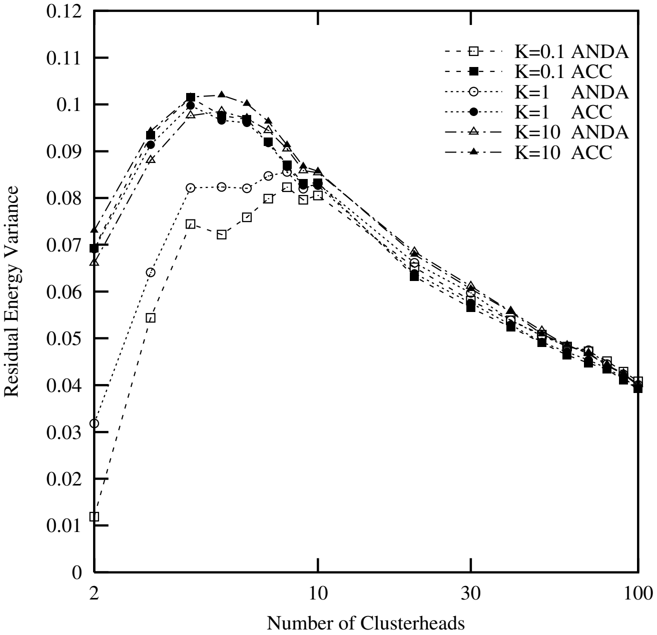


Fig. 4. Static scenario: Variance of the residual energy at the cluster-heads as a function of the number of cluster-heads. Curves are plotted for a number of nodes equal to 1000 and for varying values of K. Results obtained through ANDA and the ACC scheme are compared.

equal distribution of the nodes among the clusters. For any value of K ANDA outperforms the ACC scheme.

Next, we consider the dynamic scenario with $C = 100$ and $N = 1000$. In this case, periodical updates of the network configuration are executed; the more frequently the network configuration is updated, the greater the network life-time and the system complexity. Thus, results showing the trade-off between network life-time and number of executed configuration updates are presented.

Fig. 5 presents the network life-time for different values of K and nodes uniformly distributed in the network area. In abscissa, it is reported the number of performed configuration updates normalized to the observation time expressed in time steps. The life-time significantly increases as the number of reconfigurations grows since the energy available in the system is better exploited. For all values of K and a normalized number of updates equal to 1, an improvement of about 50% with respect to the case where ANDA is applied to the static scenario is obtained.

Finally, we expect that by combining the proposed assignment scheme with cluster-heads rotation [2], the network life-time will further increase. However, cluster-heads rotation involves an election procedure during which all nodes must be synchronized, thus resulting in an increased system complexity as well.

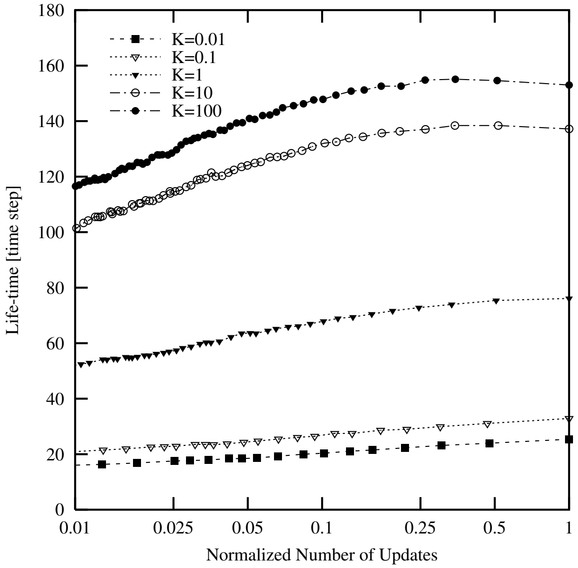


Fig. 5. Dynamic scenario: Life-time versus the normalized number of configuration updates, for a number of nodes equal to 1000, for a number of cluster-heads equal to 100 and different values of K. Nodes are uniformly distributed in the network area.

5 Conclusions

We addressed the problem of maximizing the life-time of a wireless ad hoc network, i.e., the time period during which the network is fully working. We focused on cluster-based networks and presented an original solution that maximizes the network life-time by determining the optimal clusters size and assignment of nodes to cluster-heads. We considered two working scenarios: in the former, the network configuration is computed only once; in the latter, the network configuration can be periodically updated. We obtained improvements in the network life-time equal to 15% in the case of the static scenario, and up to 74% in the case of the dynamic scenario.

References

1. Chang, J.H., Tassiulas, L., "Energy conserving routing in wireless ad hoc networks," *Proc. INFOCOM 2000,* Tel-Aviv, Israel, Mar. 2000.
2. Rabiner Heinzelman, W., Chandrakasan, A., Balakrishnan, H., "Energy-Efficient Communication Protocols for Wireless Microsensor Networks," *Proc. HICSS,* Maui, Hawaii, Jan. 2000.
3. Haker, D.J., Ephremides, A., "The Architectural Organization of a Mobile Radio Network via a Distributed Algorithm," *IEEE Trans. on Comm.,* Vol. 29, No. 11, Nov. 1981, 1694–1701.

4. Gerla, M., Tsai, T.-C., “Multicluster, Mobile, Multimedia Radio Network,” *ACM/Baltzer Journal of Wireless Networks,* Vol. 1 , No. 3, 1995, 255–265.
5. Kwon, T., Gerla, M., “Clustering with Power Control,” *Proc. MILCOM’99,* Nov. 1999.
6. Lin, C.R., Gerla, M., “Adaptive Clustering for Mobile Wireless Networks,” *IEEE JSAC,* Vol. 15, No. 7, Sep. 1997, pp. 1265–1275.
7. Basagni, S., “Distributed and Mobility-Adaptive Clustering for Multimedia Support in Multi-Hop Wireless Networks,” *IEEE VTC’99,* 889–893.
8. McDonald, A.B., “A Mobility-Based Framework for Adaptive Clustering in Wireless Ad-Hoc Networks,” *IEEE JSAC,* Vol. 17, No. 8, Aug. 1999, 1466–1487.
9. Pearlman, M.R., Haas, Z.J., “Determining the Optimal Configuration for the Zone Routing Protocol,” *IEEE JSAC,* Vol. 17, No. 8, Aug. 1999, 1395-1414.

Performance of Multipoint Relaying in Ad Hoc Mobile Routing Protocols

Philippe Jacquet, Anis Laouiti, Pascale Minet, and Laurent Viennot

INRIA
Domaine de Voluceau
78153 Le Chesnay cedex
France
{philippe.jacquet, anis.laouiti, pascale.minet, laurent.viennot}@inria.fr

Abstract. We analyze the performance of *ad hoc* pro-active routing protocols. In particular we focuse on the multipoint relay concept introduced in OLSR protocol and which brings a significant improvement in broadcast control traffic overhead.We analyze the performance in two radio network model: the random graph model and the unit graph model. The random graph is more suitable for the modelization of indoor networks. The unit graph is more suitable for outdoor networks. We compare the performance of OLSR with the performance of basic link state protocols using full flooding.

1 Introduction

Radio networking is emerging as one of the most promising challenge made possible by new technology trends. Mobile Wireless networking brings a new dimension of freedom in internet connectivity. Among the numerous architectures that can be adapted to radio networks, the *Ad hoc* topology is the most attractive since it consists to connect mobile nodes without pre-existing infrastructure. When some nodes are not directly in range of each other there is a need of packet relaying by intermediate nodes. The working group MANet of Internet Engineering Task Force (IETF) is standardizing routing protocol for *ad hoc* wireless networking under Internet Protocol (IP). In MANet every node is a potential router for other nodes. The task of specifying a routing protocol for a mobile wireless network is not a trivial one. The main problem encounterd in mobile networking is the limited bandwidth and the high rate of topological changes and link failure caused by node movement. In this case the classical routing protocol as Routing Internet Protocol (RIP) and Open Shortest Path First (OSPF) first introduced in ARPANET [1] are not adapted since they generate too much control traffic and can only accept few topology changes per minute.

MANet working group proposes two kinds of routing protocols: the reactive protocols and the pro-active protocols. The reactive protocols such as AODV [3], DSR [2], and TORA [4], do not need control exchange data in absence of data traffic. Route discovery procedure is invoked on demand when a source has a new connection pending toward a new destination. The route discovery procedure in

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 387–398, 2002.

general consists into the flooding of a *query* packet and the return of the route by the destination. The exhaustive flooding can be very expensive, thus creating delays in route establishment. Furthermore the route discovery via flooding does not guarantee to create optimal routes in terms of hop-distance.

The pro-active protocols such as Optimized Link State Routing (OLSR) [5], TBRPF [6], need periodic update with control packet and therefore generates an extra traffic which adds to the actual data traffic. The control traffic is broadcasted all over the network via optimized flooding. Optimized flooding is possible since nodes permanently monitor the topology of the network. OLSR uses multipoint relay flooding which very significantly reduce the cost of such broadcasts. Furthermore, the node have permanent dynamic database which make optimal routes immediately available on demand. The protocol OLSR has been adapted from the *intra-forwarding* protocol in HIPERLAN type 1 standard [7]. Most of the salient features of OLSR such as multipoint relays and link state routing are already existing in the HIPERLAN standard.

The aim of the present paper is to analyze the performance of the multipoint relaying concept of OLSR under two models of network: the random graph model and the unit graph model. The paper is divided into four main sections. The use of analytical models is very interesting because it captures the essential of the algorithms that cannot be captured by simulations because of the combinatorics explosion of parameters to tune. This follows the first attempts of ad hoc routing analytical modeling in [15,16]. The first section summarizes the main feature of OLSR protocol. The second section introduces and discusses the graph models. The third section develops the performance analysis of OLSR with respect to the graph models. We give few theoretical theorems about the performance of multipoint relaying without proof due to the lack of place but with due reference to reports and article containing *in extenso* the proofs (mostly in the research report [14]).

2 The Optimized Link State Routing Protocol

2.1 Non Optimized Link State Algorithm

Before introducing the optimized link state routing we make a brief reminder about non optmized link state such as OSPF. In an *ad hoc* network, we call link, a pair of two nodes which can hear each other. In order to achieve unicast transmission, it is important here to use bidirectionnal link (IEEE 802.11 radio LAN standard requires a two way packet transmission). However due to sensitivity of power discrepancies, unidirectional links can arise in the network. The use of unidirectional links is possible but require different protocols and is omitted here. Each link in the graph is a potential hop for routing packets. The aim of a link state protocol is that each node has sufficient knowledge about the existing link in the network in order to compute the shortest path to any remote node.

Each node operating in a link state protocol performs the two following tasks:

- **Neighbor discovery:** to detect the adjacent links;
- **Topology broadcast:** to advertize in the whole network about important adjacent links.

By important adjacent links we mean a subset of adjacent links that permit the computation of the shortest path to any destination.

The simplest neighbor discovery consists for each node to periodically broadcast full hello packets. Each full hello packet contains the list of the heared neighbor by the node. The transmission of hello packets is limited to one hop. By comparing the list of heared nodes each host determine the set of adjacent bidirectional links.

A non optimized link state algorithm performs topology broadcast simply by periodically flooding the whole network with a topology control packet containing the list of all its neighbor nodes (i.e. the heads of its adjacent links). In other words, all adjacent links of a node are important. By flooding we mean that every node in the network re-broadcast the topology control packet upon reception. Using sequence number prevents the topology control packet to be retransmitted several times by the same node. The number of transmissions of a topology control packet is exactly N, when N is the total number of in the network, and when retransmission and packet reception are error free.

If h is the rate of hello transmission per node and τ the rate of topology control generation, then the actual control overhead in terms of packet transmitted of OSPF is

$$hN + \tau N^2 \tag{1}$$

In terms of bytes transmitted, more precisely in IP addresses unit, the overhead is

$$hNM + \tau N^2 M \tag{2}$$

where M is the average number of adjacent links per node. If M is of the same order than N then the overhead is cubic in N. Notice that the topology broadcast overhead is one order of magnitude larger than the neighbor discovery overhead.

Notice that for non-optimized link state routing the hello and topology control packet can be the same.

2.2 OLSR and MultiPoint Relay Nodes

The Optimized Link State Routing protocol is a link state protocol which optimizes the control overhead via two means:

1. the important adjacent links are limited to MPR nodes;
2. the flooding of topology control packet is limited to MPR nodes (MPR flooding).

The concept of MultiPoint Relay (MPR) nodes has been introduced in [7]. By MPR set we mean a subset of the neighbor nodes of a host which covers the two-hop neighborhood of the host. The smallest will be the MPR set the more efficient will be the optimization. We give a more precise definition of the multipoint relay set of a given node A in the graph. We define the neighborhood of A as the set of nodes which have an adjacent link to A. We define the two-hop neighborhood of A as the set of nodes which have an invalid link to A but have a valid link to the neighborhood of A. This information about two-hop neighborhood and two-hop links are made available in hello packets, since every neighbor of A periodically broadcasts their adjacent links. The multipoint relay set of A (MPR(A)) is a subset of the neighborhood of A which satisfies the following condition: every node in the two-hop neighborhood of A must have a valid link toward MPR(A).

The smaller is the Multipoint Relay set is, the more optimal is the routing protocol. [13] gives an analysis and examples about multipoint relay search algorithms. The MPR flooding can be used for any kind of long hole broadcast transmission and follows the following rule:

> *A node retransmits a broadcast packet only if it has received its first copy from a node for which it is a multipoint relay.*

Reference [7] gives a proof that such flooding protocol (selective flooding) eventually reaches all destinations in the graph.[7] also gives a proof that for each destination in the network, the subgraph made of all MPR links in the network and all adjacent links to host A contains a shortest path with respect to the original graph.

Therefore the multipoint relays improve routing performance in two aspects: first it significantly reduces the number of retransmissions in a flooding or broadcast procedure; second it reduces the size of the control packets since OLSR nodes only broadcast its multipoint relay list instead of its whole neighborhood list in a plain link state routing algorithm.

In other words if D_N is the average number of MPR links per node and R_N the average number of retransmission in an MPR flooding, then the control traffic of OLSR is, in packet transmitted:

$$hN + \tau R_N N \ , \tag{3}$$

and, in IP addresses transmitted:

$$hMN + \tau R_N D_N N \ . \tag{4}$$

Notice that when the nodes selects all their adjacent links as MPR links, we have $D_N = M$ and $R_N = N$: we have the overhead of a full link state algorithm. However we will show that straightforward optimizations make $D_N \ll M$ and $R_N \ll N$ gaining several orders of magnitude in topology broadcast overhead. Notice that the neighbor discovery overhead is unchanged. Summing both overhead we may expect that OLSR has an overhead reduced of an magnitude order with respect to full link state protocol.

The protocol as it is proposed in IETF may differ to some details from this very simple presentation. The reason is for second order optimization with regards to mobility for example. For example hosts in actual OLSR do not advertize their MPR set but their MPR selector set, i.e. the subset of neighbor nodes which have selected this host as MPR.

2.3 MPR Selection

Finding the optimal MPR set is an NP problem as proven in [8]. However there are very efficient heuristics. Amir Qayyum [13] has proposed the following one:

1. select as MPR, the neighbor node which has the largest number of links in the two-hop neighbor set;
2. remove this MPR node from the neighbor set and the neighbor nodes of this MPR node from the two-hop neighbor set;
3. the previous steps until the two-hop neighbor set is empty.

An ultimate refinement is a prior operation which consists into detecting in the two-hop neighbor the node which have a single parent in the neighbor set. These parents are selected as MPR and are eliminated from the neighbor set, and their neighbor are eliminated in the two-hop neighbor set.

It is proven in [8,10] that this heuristic is optimal by a factor $\log M$ where M is the size of the neighbor set (i.e. the heuristical MPR set is at most $\log M$ times larger than the optimal MPR set).

3 The Graph Models

The modelization of ad hoc mobile network is not an easy task. Indeed the versatility of radio propagation in presence of obstacles, distance attenuation and mobility is the source of incommensurable difficulties. In passing one should notice that mobility not only encompass host mobility but also the mobility of the propagation medium. For example when a door is open in a building, then the distribution of links change. If a truck passes between two host it may switch down the link between them. In this perspective building a realistic model that is tractable by analysis is hopeless. Therefore we will focuse to build models dedicated to specific scenarios.

There are two kinds of scenarios: the indoor scenarios and the outdoor scenarios. For the indoor scenario we will use the random graph model. For the outdoor scenarios we will use the random cartesian graph model. The most realistic model lies somewhere between the random graph model and the random cartesian graph model.

3.1 The Random Graph Model for Indoor Networks

In the following we consider a wireless indoor network made of N nodes. The links are distributed according to a random graph with N vertices an link probability p. In other words, a link exists between two given nodes with probability p. Link's existence are independent from one pair of nodes to another. Figure 1 shows an example of a random graph with $(N, p) = (10, 0.7)$, the nodes have been drawed in concentric mode just for convenience.

The random graph model implicitly acknowledge the fact that in an indoor network, the main cause of link obstruction is the existence of random obstacle (wall, furniture) between any pair of nodes. The fact that the links are independently distributed between node pairs assumes that these obstacles are independly distributed with respect to node position, which of course is never completely true. However the random graph model is the simplest satisfactory model of indoor radio network and provides excellent results as a starting point.

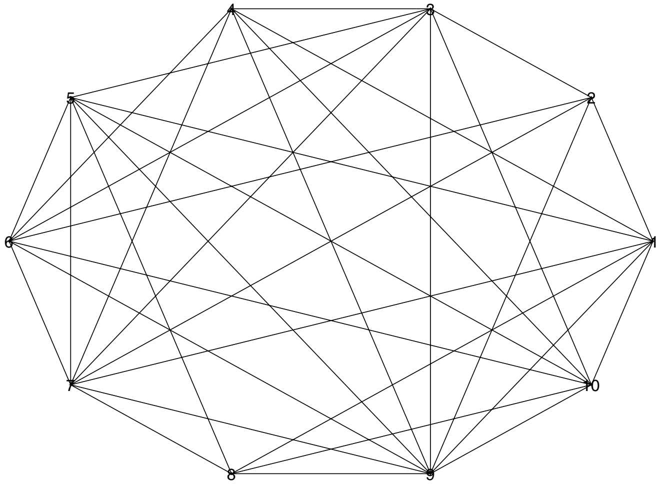


Fig. 1. A random graph with $n = 10$ and $p = 0.7$, generated by Maple

When the network is static, then the graph does not change during the time. It is clear that nodes does not frequently change position in indoor model, but the propagation medium can vary. In this case the random graph may vary with the time. One easy way to model time variation is to assume random and independent link lifetime. For example, one can define μ as link variation rate, i.e. the rate at which each link may come down or up. During an interval $[t, t + dt]$ a link can change its status with probability μdt, i.e. it takes status "up" or "down" with probability p, independently of its previous status. The effect of mobility won't be investigated in the present paper.

3.2 The Random Unit Graph Model for Outdoors Networks

To explain this kind of graph it suffices to refer to a very simple example. Let L be a non-negative number and let us define a two-dimensional square of size $L \times L$ unit lengths. Let consider N nodes uniformly distributed on this square. The unit graph is the graph obtained by systematically linking pairs nodes when their distance is smaller or equal to the unit length. This model of graph is well adapted to outdoor networks where the main cause of link failure is the attenuation of signal by the distance. In this case the area where a link can be established with a given host is exactly the disk of radius the radio range centered on the host. However the presence of obstacle may give a more twisted shape to the reception area (that may not be single connected).

Figure 2 shows a random unit graph of dimension two. The random unit graph is built in two step: the first step is the uniform distribution of the point on the rectangle area; the second step is the link distribution between node pair according to their distance.

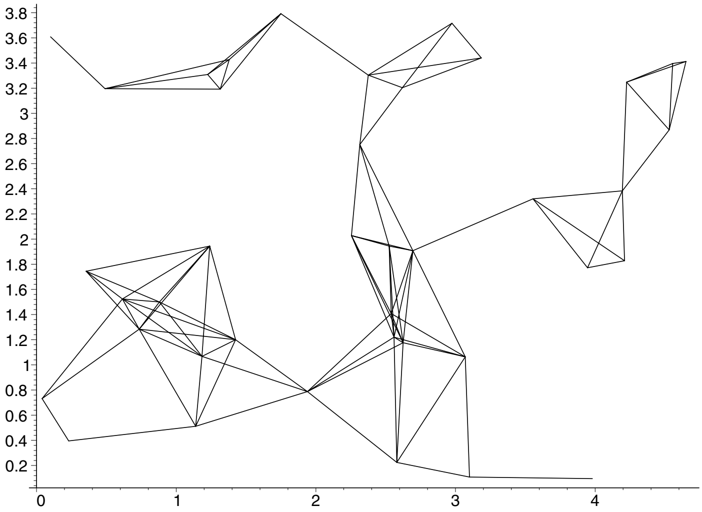


Fig. 2. The random Unit graph derived from random forty points locations in a 5 × 4 rectangle

The reception area may also change with the time, due to node mobility, obstacle mobility, noise or actual data traffic. In the present paper we will assume that the network is static.

Of course, the unit graph can be defined on other space than the plane. For example a unit graph can be defined on a 1D segment, modeling a mobile network made of cars on a road. It can be a cube in the air, modeling a mobile network made of airplanes, for example.

4 Analysis of OLSR in the Random Graph Model

The proof of the results listed in this section can be found in [9].

Most pro-active protocols (like OLSR) have the advantage to deliver optimal routes (in term of hop number) to data transfers (the proof of this is in [7] and [14]). The analysis of optimal routes is very easy in random graph models since a random graph tends to be of diameter 2 when N tends to infinity with fixed p.

Theorem 1. *The optimal route between two random nodes in a random graph, when N tends to infinity, is either of length 1 with probability p,or of length 2 with probability $q = 1 - p$.*

Theorem 2. *For all $\varepsilon > 0$, the optimal MPR set size D_N of any arbitrary node is smaller than $(1+\varepsilon)\frac{\log N}{-\log q}$ with probability tending to 1 when N tends to infinity.*

Notice that $D_N = O(\log N)$ which very favorably compares to the size of the the whole host neighborhood which (in average pN) and considerably reduces the topology broadcast.

Theorem 3. *The broadcast or flooding via multipoint relays takes in average a number R_N of retransmissions smaller than* $(1+\varepsilon)\frac{\log N}{-p\log q}$.

Corollary 1. *The cost of OLSR control traffic for topology broadcast in the random graph model is $O(N(\log N)^2)$ compared to $O(N^3)$ with plain link state algorithm.*

Remark: Notice that the neighbor sensing in $O(N^2)$ is now the dominant source of control traffic overhead.

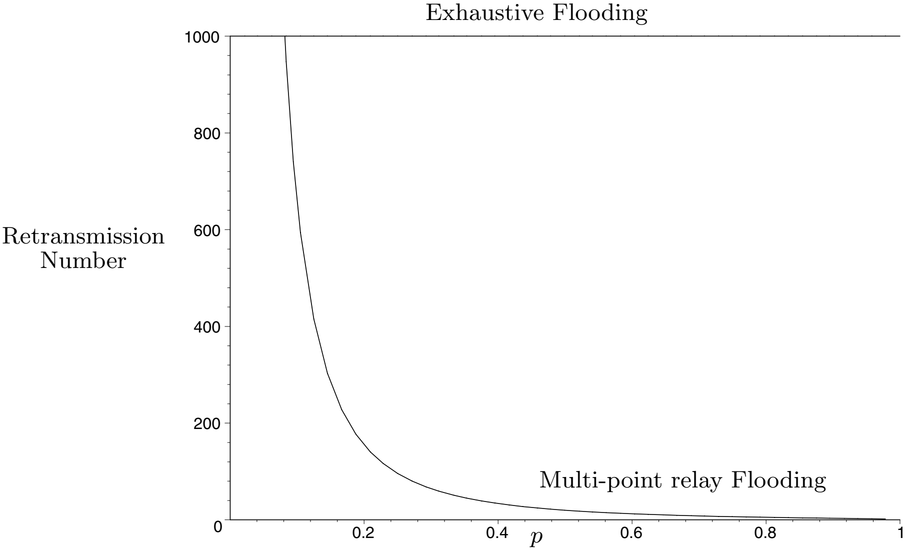


Fig. 3. average number of retransmissions in multi-point relay flooding with $N = 1000$ and p variable

5 Analysis of OLSR in the Random Unit Graph

5.1 Results in 1D and 2D Random Unit Graphs

We present results for 1D and 2D random unit graphs. The proof of the results shown in this section can be found in [14]. A 1D Unit graph can be made of N nodes uniformly distributed on a strip of land whose width is smaller than the radio range (set as unit length). We assume that the length of the land strip is L unit length.

Theorem 4. *The size of the MPR set D_N of a given host is 1 when the host is at less than one radio hop to the end to one end of the strip, and 2 otherwise.*

Theorem 5. *The MPR flooding of a broadcast message originated by a random node takes $R_N = \lfloor L \rfloor$ retransmission of the message when N tends to infinity and L is fixed.*

Notice this is assuming an error free retransmission. In case of error, the retransmission stops at the first MPR which does not receive correctly the message. In order to cope with this problem one may have to add redundance in the MPR set which might be too small with regard to this problem.

Notice that these figures favorably compare with plain link state where $D_N = M = N/L$ and $R_N = N$.

The analysis in 2D is more interesting because it gives less trivial results.

Theorem 6. *When L is fixed and N increases, then the average size of the MPR set, D_N tends to be smaller than* $3\pi(N/(3L^2))^{1/3} = 3\pi(M/(3\pi))^{1/3}$.

Notice that this figure compares favorably with plain link state where $D_N = M = N/L^2$.

Figure 4 displays simulation results for dimension 2. The heuristic has been applied to the central node of a random 4×4 unit graph. The convergence in $M^{1/3}$ is clearly shown. Notice that in this very case the upper bound of D_N is at least greater by a factor 2 than actual values obtained by simulations. Figure 5 summarizes the results obtained for quantity D_N in the random graph model for dimension 1 and 2. The results for dimension 2 have been simulated.

Theorem 7. *The MPR flooding of a broadcast message originated by a random node takes $R_N = O((NL^4)^{1/3})$ retransmissions of the message when N tends to infinity and L is fixed.*

5.2 Comparison with Dominating Set Flooding

In [11] Wu and Li introduced the concept of dominating set. They introduced two kinds of dominating set that we will call, the rule 1 dominating set and the rule 2 dominating set. In this section we establish quantative comparisons between the performance of dominating set flooding and MPR flooding. In particular we will show that dominating set floodings does not outperform significantly full flooding in random graph models and in random unit graph of dimension 2 and higher. Rule 1 dominating set does not outperform significantly full flooding in random unit graph model of dimension 1. MPR flooding outperforms both dominating set flooding in any graph models studied in this paper.

The dominating set flooding consists into restricting the retransmission of a broadcast message to a subset of nodes, called the dominating set. Rule 1 and rule 2 consist into two different rules of dominating set selection. The rules consist into compairing neighbor sets (for example by checking hellos). For a node A we denote by $\mathcal{N}(A)$, the neighbor set of node A.

In rule 1, a node A does not belong to the dominating if and only if there exists a neighbor B of A such that

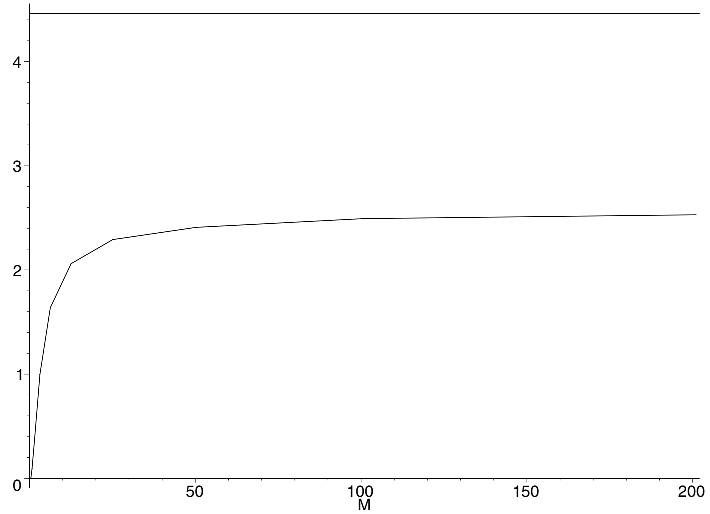


Fig. 4. Bottom: simulated quantity $D_N/M^{1/3}$ versus the number of neighbor M for the central position in a 4×4 random unit graph,top: upper bound obtained in theorems.

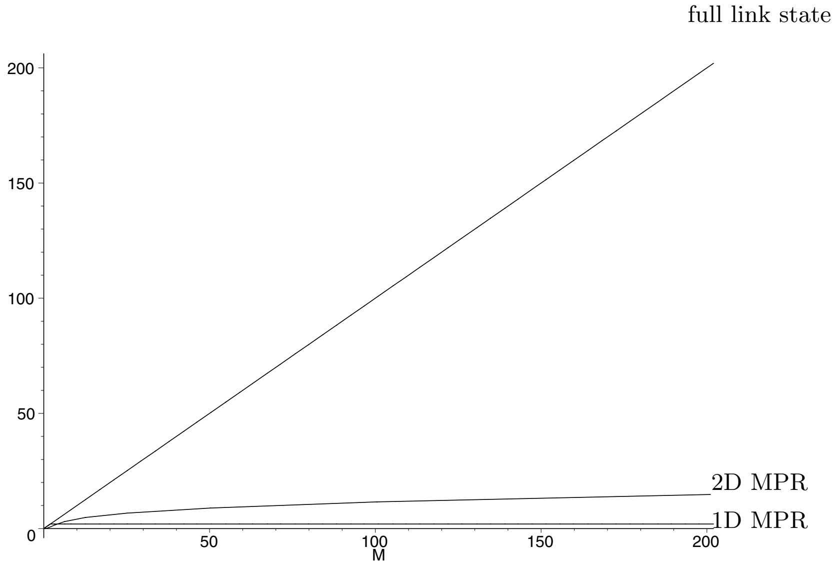


Fig. 5. Unit graph model from bottom to top, average number of MPR for 1D, 2D and full links state protocol versus the average number of neighbor nodes M.

1. B is in the dominating set;
2. the IP address of B is higher than the IP address of A;
3. $\mathcal{N}(A) \subset \mathcal{N}(B)$.

In this case one says that B dominates A in rule 1.

In rule 2, a node A does not belong to the dominating if and only if there exist two neighbor B and C of A such that

1. B and C are in the dominating set;
2. nodes B and C are neighbors;
3. the IP addresses of B and C are both higher than the IP address of A;
4. $\mathcal{N}(A) \subset \mathcal{N}(B) \cup \mathcal{N}(C)$.

In this case one says that (B, C) dominates A in rule 2.

We first, look at the performance of dominating set flooding in the random graph model (N, p).

Theorem 8. *The probability that a node in a random graph (N, p) does not belong to the dominating set is smaller than $N(1-(1-p)p)^N$ in rule 1, and smaller than $N^2(1-(1-p)^2p)^N$ in rule 2.*

Theorem 9. *In the random unit graph model of dimension 1, assuming independence between node location and node IP addresses, the probability that a node does not belong to the dominating set in rule 1 is smaller than $\frac{4}{M}$ and the average size of the dominating set in rule 2 is* $\max\{0, 2L-1\}$.

Remark: The proofs of these theorem can be found in [14]. The density of the dominating set in rule 2 is twice than the density of retransmitters in MPR flooding when the network model is the random unit graph of dimension one.

In random graph of dimension 2 and higher the probabilities that a node does not belong to the dominating set in rule 1 or in rule 2 are $O(1/M)$ since it is impossible to cover one unit disk with two unit disk that have different centers.

6 Conclusion and Further Works

We have presented a performance evaluation of OLSR mobile ad-hoc routing protocols in the random graph model and in the random unit graph model. The originality of the performance evaluation is that it is completely based on analytical methods (generating function, asymptotic expansion) and does not rely on simulation software. The random graph model is enough realistic for indoor or short range outdoor networks where link fading mainly comes from random obstacles. The random unit graph model is realistic for long range outdoor networks where link fading mainly comes from distance attenuation. In this case the random graph model can be improved by letting the parameter p depending on distance x between the nodes. This will be subject of further works.

References

1. J.M. McQuillan, I. Richer, E.C. Rosen, "The new routing algorithm for the ARPANET," *IEEE Trans. Commun.* COM-28:711-719.

2. D.B. Johnson, D.A. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks," in *Mobile Computing*, Ch. 5, pp 153-181, Kluwer Academic Publisher, 1996.
3. C.E. Perkins, E.M. Royer, "Ad Hoc On-Demand Distance Vector Routing," *IEEE Workshop on Mobile Computing Systems and Applications*, pp. 90-100, 1999.
4. M.S. Corson, V. Park, "Temporallly ordered routing algorithm," draft-ietf-manet-tora-spec-02.txt, 1999.
5. P. Jacquet, P. Muhlethaler, A. Qayyum, A. Laouiti, L. Viennot, T. Clausen, MANET draft "draft-ietf-manet-olsr-02.txt," 2000.
6. B. Bellur, R. Ogier, F. Templin, "Topology broadcast based on reverse-path forwarding," draft-ietf-manet-tbrpf-01.txt, 2001.
7. P. Jacquet, P. Minet, P. Muhlethaler, N. Rivierre, "Increasing reliability in cable-free Radio LANs: Low level forwarding in HIPERLAN," in *Wireless Personal Communications* Vol 4, No 1, pp. 51-63, 1997.
8. L. Viennot, "Complexity results on election of multipoint relays in wireless networks," INRIA RR-3584, 1998. http://www.inria.fr/rrrt/rr-3584.html
9. P. Jacquet, A. Laouiti, "Analysis of mobile ad hoc network routing protocols in random graphs," INRIA RR-3835, 1999. http://www.inria.fr/rrrt/rr-3835.html
10. A. Qayyum, L. Viennot, A. Laouiti, "Multipoint relaying: An efficient technique for flooding in mobile wireless networks," INRIA RR-3898, 2000. http://www.inria.fr/rrrt/rr-3898.html
11. J. Wu, H. Li, "On calculating connected dominating set for efficient routing in ad hoc wireless networks," in Proc. DIAL M, 1999.
12. P. Jacquet, L. Viennot, "Overhead in mobile ad hoc network protocols," INRIA RR-3965, 2000.http://www.inria.fr/rrrt/rr-3965.html
13. A. Qayyum, *Analysis and evaluation of channel access schemes and routing protocols for wireless networks*, Thèse de l'Université Paris 11, 2000.
14. P. Jacquet, A. Laouiti, P. Minet, L. Viennot, "Performance analysis of OLSR multipoint relay flooding in two ad hoc wireless network models," INRIA Research Repport RR-4260, 2001. http://www.inria.fr/rrrt/rr-4260.html
15. A. D. Aron *et al.*, "Analytical comparison of local and end-to-end error recovery in reactive routing protocols for MANET," 3rd ACM MSWiM 2000.
16. A. Boukerche *et al.*, "Analysis of randomized congestion control with DSDV routing in ad hoc wireless networks," in JPDC, pp. 967-995, Vo 61, 2001.

An Adaptive Location-Aware MAC Protocol for Multichannel Multihop Ad-Hoc Networks

Zi-Tsan Chou[1,2], Ching-Chi Hsu[1,3], and Ferng-Ching Lin[1,2]

[1] Department of Computer Science and Information Engineering
National Taiwan University, Taipei, 106, Taiwan
{d5526005, cchsu, fc_lin}@csie.ntu.edu.tw
[2] Institute for Information Industry, Taipei, 106, Taiwan
{ztchou, fclin}@iii.org.tw
[3] Kai Nan University, Tauyan, Taiwan

Abstract. In a multihop MANET (mobile ad-hoc network), reliable broadcast support at the MAC layer will be of great benefit to the routing function, multicasting applications, cluster maintenance, and real-time systems. In this paper, we propose a new hybrid MAC protocol, called the *adaptive location-aware broadcast* (ALAB) protocol, for link-level broadcast support in multichannel systems. ALAB is scalable and mobility-transparent since it does not require any link state information. Above all, in ALAB, both deadlock and hidden terminal problems are completely solved. In principle, ALAB tries to combine both of the advantages of the allocation- and contention-based protocols and overcomes their individual drawbacks. At high traffic or density, ALAB outperforms the pure TDMA because of spatial reuse and dynamic slot management. At low traffic or density, ALAB outperforms the pure CSMA/CA because of its embedded stable tree-splitting algorithms. In addition, ALAB provides deterministic access delay bounds from its base TDMA allocation protocol. Simulation results do confirm the advantage of our scheme over other MAC protocols, such as IEEE 802.11, ADAPT, and ABROAD, even under the fixed-total-bandwidth model.

1 Introduction

With the revolutionary advances of wireless technology, the applications of the MANET (mobile ad-hoc network) are getting more and more important, especially in the emergency, military, and outdoor business environments, in which instant fixed infrastructure or centralized administration is difficult or too expensive to establish. In the MANET, pair of nodes communicates by sending packets either over a direct wireless link or through a sequences of wireless links including some intermediate nodes. Due to the broadcast nature of the radio transmission medium and the rapidly dynamic topology changes in the MANET, every algorithms and protocols developed on it will face great challenges. In this paper, we are specially interested in a *medium access control* (MAC) protocol for multihop ad-hoc networks with multiple frequency channels.

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 399–410, 2002.

A MAC protocol is to address how to allocate the multiaccess medium and resolve potential contention/collision among various nodes. MAC protocols proposed so far can be approximately classified into two categories [5]. One is allocation-based protocols, and the other is contention-based protocols. Deterministic allocation-based protocols, such as TDMA and its variants [9], are primarily designed to support bounded delay topology-independent transmissions by scheduled slot assignments. Nevertheless, these protocols are insensitive to variations in network loads or node connectivity. Although dynamic topology-dependent TDMA-based transmission scheduling protocols [13] can adjust themselves to node connectivity, they are not suitable for highly mobility environments due to heavy loads on updated link state information maintenance. As to the contention-based protocols, such as CSMA/CA and it variants [14], they are primarily designed to support asynchronous transmissions and burst traffic. However, CSMA/CA is inherently unstable [7]. Because of this reason, the CARMA protocol based on the deterministic tree-splitting algorithm [7] was proposed. In CARMA, in order to maintain a consistent channel view for all nodes in a multihop wireless network, a base station should be set up to govern this task. Hence it is not suitable for the large-scale MANET.

Most previous works on MAC protocols including IEEE 802.11, ADAPT [4], CARMA [7], and GRID [14] are designed to support only reliable unicast transmission. As indicated in [5,8,11], support for reliable broadcast at the MAC layer will be of great benefit to the routing function, multicasting applications, cluster maintenance, and real-time systems. Clearly, a single reliable broadcast can be implemented by sending one or more reliable unicast messages. However, this approach is not scalable since the time to complete a broadcast increases with the number of neighbors. Besides, MAC protocols typically do not maintain link state information [5]. Recently, several MAC protocols for broadcast support have been proposed, including ABROAD [5], TPMA [8], RBRP [11], CATA [13], and FPRP [15]. All of them depend on the collision detection capacity. In TPMA and RBRP, nodes with bad luck in their elimination phase or reservation request phase may lead to starvation. To make matters worse, all of these protocols may lead to deadlocks. A *deadlock* [11] is said to occur if two conflicting broadcasts are scheduled in the same slot and the senders do not realize this conflict. We also notice that all the above-mentioned protocols have focused only on single channel systems. From many literatures [9,14], we know that a multichannel system outperforms a single channel system in many aspects, including throughput, reliability, bandwidth utilization, network scalability, synchronization implementation, and QoS support.

The authors in [5] developed a novel hybrid MAC protocol, called ABROAD, for reliable broadcast in single channel MANETs. Importantly, they try to combine both of the advantages of the allocation- and contention-based protocols and overcomes their individual drawbacks. Thus, ABROAD can dynamically self-adjust its behavior according to the prevailing network conditions [4,5]. Following their hybrid approach, but with a whole different design strategy, we propose a new multichannel MAC protocol based on the tree-splitting algorithms for link-level broadcast support in multihop ad-hoc networks. We call the resulting distributed protocol "Adaptive Location-Aware Broadcast" (ALAB) protocol. Since a MANET should operate in a physical area, it is very natural to exploit location information in such an environment [14]. In addition, via a GPS (Global Positioning Systems), every node can get absolute timing and location information; thus synchronization becomes easy [8,11,12]. The advantages of the ALAB protocol are as follows. (i) ALAB supports reliable unicast, multicast, and broadcast transmission services in an integrated manner. That is, unicast and multicast packets are considered as special cases of broadcast packets. (ii) ALAB is scalable and mobility-transparent since it does not require any link state information. Moreover, both the time to broadcast a packet and the number of channels required for the MANET are independent of the network topology. (iii) In ALAB, no hidden or exposed terminal problems will exist. Therefore, our design does not need any handshake process such as RTS/CTS or RTS/CR/RA [8]. (iv) Like ABROAD, ALAB also provides bounded access delay from its base TDMA allocation protocol. Furthermore, ALAB is stable and adapts well to any traffic and network topologies. (v) In ALAB, the deadlock and starvation problems are completely eliminated. (vi) Under the severe traffic load and node density conditions, ALAB delivers superior performance than ABROAD, which outperforms TDMA, IEEE 802.11, and ADAPT [4,5], even under the fixed-total-bandwidth model.

2 The ALAB Protocol

2.1 Model and Assumptions

A multihop mobile radio network used to pass messages containing data and control information can be modelled as an undirected graph $G = (\mathcal{V}, \mathcal{E})$ in which $\mathcal{V}$ ($|\mathcal{V}| = N$) is the set of mobile hosts and there is an edge $(u, v) \in \mathcal{E}$ if and only if u and v can mutually receive each other's transmissions. In this case, we say that u and v are neighbors. Note that the edge set may *vary* over time because of nodal mobility. We can assign each node v in the network a unique identifer (ID) by a number in $\aleph = \{0, 1, \ldots, N-1\}$, where $|\aleph| = N$. In this paper, all logarithms are assumed to be base 2. Given an integer $v \in \aleph$, let $Binary(v) = (v_1 v_2 \ldots v_{k-1} v_k)$ denote its binary string, where $k = \lceil \log N \rceil$. Thus, every integer in $\aleph$ can be represented by a unique binary k-tuples $(v_1 v_2 \ldots v_{k-1} v_k)$, where $v_i \in \{0, 1\}$. In addition, each channel is uniquely assigned by a number in $\mathcal{C} = \{0, 1, \ldots, \rho - 1\}$, where $1 \leq \rho < N$.

Within a TDMA network, the time axis is divided into units called (transmission) *frames*, and each frame is composed of time slots. Each slot in turn comprises mini-slots. Nodes in the network are assumed to be synchronized and that the frame length is the same for each node. Each mobile radio host in a multichannel network is equipped with the transceivers (a single transmitter and multiple receivers). Depending on the ability of the transceivers, each node can communicate with others either in the full-duplex mode or in the half-duplex mode. In the half-duplex mode, each host cannot transmit and receive at the

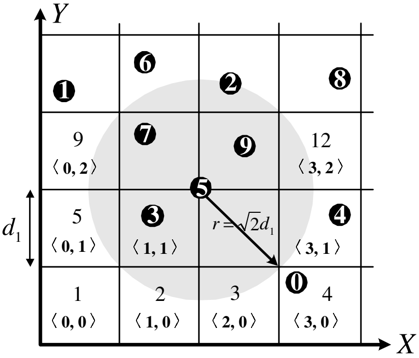


Fig. 1. Ten mobile hosts are dispersed randomly over the 2D geographic region. The integer within in the grid is the channel number, while the integer pairs are the grid coordinate. The center part of the geographic area shows the relation between r and d.

same time [3]. In the full-duplex mode, each host can transmit only one packet on one channel but receive multiple packets on all channels simultaneously [9]. Throughout this paper, we assume every node works in the full-duplex mode. On the same channel, two types of communication collisions will arise [9]. The primary collision occurs when a node transmitting in a given mini-slot is receiving in the same mini-slot on the same channel. This also implies the converse: a receiving node cannot be transmitting on the same channel at the same time. The secondary collision occurs when node receives more than one packet in a mini-slot on the same channel. In both cases, all packets are rendered useless. To this end, we assume that if more than one node is transmitting on the same channel such that the packets overlap in time, then collision occurs on that channel. On the other hand, simultaneous reception of packets on other channels is not affected [3,9]. In this paper, we also assume that a node is capable of determining the current status of a single radio channel [12]. That is, at the end of a mini-slot, each node can obtain feedback from the receiver specifying whether the status of a radio channel is (i) NULL: no transmission on the channel, (ii) SINGLE: exactly one transmission on the channel, or (iii) COLLISION: two or more transmissions on the channel.

The basic idea behind the ALAB protocol is very simple; in brief, we just imitate the organization of cellular/cluster networks. This approach is widely adopted in many issues for the MANET [6,14]. Each node is assumed to know its own position by virtue of its GPS receiver but not the position of any other nodes in the network. In our model of the ad-hoc network, nodes are dispersed randomly over a pre-defined geographic region, which is partitioned into two-dimensional logical grids as illustrated in Fig. 1. Each grid is a square of size $d \times d$. Let r_i be the transmission range of node i. Determining the optimal values of r_i and d is not an easy task. In our design, we restrict $\sqrt{2}d \leq r_i \leq 2d$. Let $\aleph_\infty = \{0, 1, 2, 3, \ldots\}$. Grids are numbered $\langle x, y \rangle$ following the conventional xy-coordinate. Every node must know how to map its physical location $(x', y') \in$

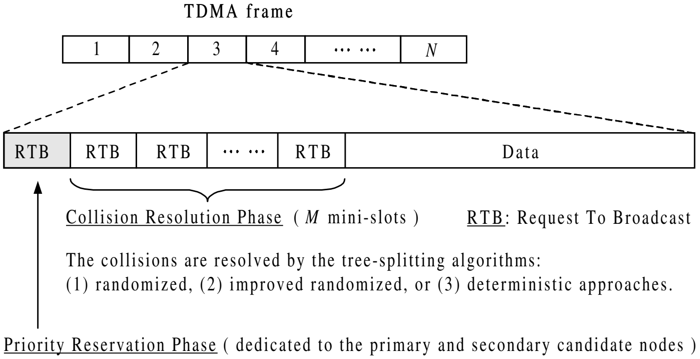

TDMA frame
1
2
3
4
⋯⋯
N
RTB
RTB
RTB
⋯⋯
RTB
Data
Collision Resolution Phase (M mini-slots)
RTB: Request To Broadcast
The collisions are resolved by the tree-splitting algorithms:
(1) randomized, (2) improved randomized, or (3) deterministic approaches.
Priority Reservation Phase (dedicated to the primary and secondary candidate nodes)

Fig. 2. The ALAB slot and frame structure.

$(\Re, \Re)$ to the corresponding grid coordinate $\langle x, y \rangle \in \langle \aleph_\infty, \aleph_\infty \rangle$. As illustrated in Fig. 1, we assign $\langle x, y \rangle = \langle \lfloor \frac{x'}{d} \rfloor, \lfloor \frac{y'}{d} \rfloor \rangle$. Besides, each grid is assigned a unique channel. When a node is located at a grid $\langle x, y \rangle$, it must use the channel assigned to the grid $\langle x, y \rangle$ for transmission. Given two different girds $\langle x_1, y_1 \rangle$ and $\langle x_2, y_2 \rangle$, if $\max\{|x_1 - x_2|, |y_1 - y_2|\} \leq 4$, then these two girds are called the *interfering grids*. The interfering grids are forbidden to be assigned the same channel to prevent co-channel interference in the packet transmission phase of ALAB. (We will explain it in the next subsection.) To attain this goal, we can simply apply the distance-4 coloring algorithms [14] to assign a channel for each grid. In the meantime, the frequency reuse should be maximized. Fig. 1 shows the possible channel assignments. Let $|\mathcal{G}|$ be the total number of grids over the geographic region. By a simple counting, the total number of channels required for the ALAB protocol is $\min\{25, |\mathcal{G}|\}$.

The main purposes of these restrictions are as follows. Nodes within the same grid form a *single-hop* cluster. In other words, all nodes within the same grid can hear the transmission of others. By the collision detection ability of the transceivers, all nodes within the same grid are able to maintain a consistent channel view. Due to the channel consistency in every grid, no deadlock or hidden terminal problems will exist.

2.2 Protocol Description

The ALAB protocol integrates a tree-splitting collision resolution protocol within each slot of a TDMA allocation protocol. Each node is assigned a transmission schedule (frame) consisting of N slots. The slot and frame structure of the ALAB protocol, which is somewhat like the design of HYPERLAN [1], is shown in Fig. 2. The frame is divided into fixed-sized slots. Each slot is composed of three parts: a *priority reservation* phase and a *collision resolution* phase followed by a *packet transmission* phase. The first two phases are called the *leader election* phase. The priority reservation phase occupies the first mini-slot and the collision resolution phase consists of the next M mini-slots. The final mini-slot is the packet transmission phase. In the priority reservation phase, only the predetermined primary and secondary candidate nodes have the chance to reserve the slot. However, when the first mini-slot remains unused, all active nodes contend to use it either by the randomized collision resolution algorithm or by the deterministic one. A node is said an *active* node if it has packets to send. Through the leader election phase, we guarantee that at most one active node will survive in a gird. The survival(s) gets the right of broadcast in the packet transmission phase. Recall that the transmission range is limited and simultaneous reception of packets on other channels is not affected. As a result, inter-grid communications via data packets are collision-free. In the following, we will focus the protocol description on *a single grid*, say $\langle x, y\rangle$. Before we describe the ALAB protocol in detail, we also make the following assumptions. (i) A node located in a grid $\langle x, y\rangle$ is assumed to continuously monitor the status of the channel assigned to $\langle x, y\rangle$. (ii) A node wishing to transmit a packet which may has arrived in the interim, it would wait until the beginning of the next slot. (iii) The channel introduces no errors, so control-packet collisions are the only source of errors. Nodes can perfectly detect such collisions immediately at the end of each mini-slot. (iv) Each mini-slot is designed to accommodate the packet transmission time and the guard time, which corresponds to the maximum differential propagation delay between any pair of nodes. Since the network are assumed to be synchronized, all active nodes enter the priority reservation phase synchronously.

1) *Priority Reservation Phase*: In slot i of a frame, we let the node with the $ID = i - 1 \bmod N$ be the *primary candidate* (PC for short) node and the node with the $ID = i - 1 + \lfloor \frac{N}{2} \rfloor \bmod N$ be the *secondary candidate* (SC for short) node. In our design, the priority of the PC node is higher than that of the SC node. At the beginning of the first mini-slot, only the PC and SC nodes are allowed to send RTB (Request-To-Broadcast) control packets with probability 1. At the end of the first mini-slot, if the status of the channel is COLLISION, then the PC node overwhelmingly wins the slot to broadcast a packet. If the status of the channel is SINGLE, all active nodes except the winner quit the contention at the remaining mini-slots, abandon the corresponding packet transmission mini-slot and wait for the next slot. Otherwise, all active nodes enter the collision resolution phase.

2.a) *Randomized Collision Resolution Phase*: At the beginning of the ith mini-slot, where $2 \leq i \leq M + 1$, all active nodes send RTBs with probability 1. At the end of the ith mini-slot, if the status of the channel is NULL, then the collision resolution period is over. The contending nodes involved in the COLLISION split randomly into two subsets by each flipping a coin. Those who obtain heads (with probability p) send an RTB in the next mini-slot; while those who obtain tails (with probability $1 - p$) become inactive and wait for the next slot. This process keeps running until a SINGLE is reported or i equals $M + 1$, whichever comes first. The above-mentioned algorithm is similar to that of TPMA [8]. We can find that the collision resolution process stops immediately once a NULL occurs. One can make a further improvement, however. On condition that a NULL is sensed, all previous contenders are allowed to flip a coin again and those who obtain heads can send RTBs in the next mini-slot. Thus the collision resolution phase will never terminate in the NULL state before mini-slot $M+1$. It is worth noticing that nodes with bad luck in the randomized collision resolution phase will not starve because of the underlying TDMA protocol. The advantages of the randomized approach are that it achieves fairness naturally and a winner may arise quickly. A reasonable value of M could be $1+\lceil\log\frac{N}{\lceil\mathcal{G}\rceil}\rceil$.

2.b) *Deterministic Collision Resolution Phase*: Our deterministic collision resolution algorithm is similar to that in [2]. We assume that every node keeps an integer variable *temporary_ID* used for the collision resolution phase. Initially, *temporary_ID* = *ID*. We let $M = 1+\lceil\log N\rceil$ and $(b_1b_2,\cdots,b_k)$ be the binary representation of any given node *temporary_ID*, where $k = \lceil\log N\rceil$. At the beginning of the second mini-slot, all active nodes send RTB packets. If the status of the channel is NULL, then the collision resolution period is over. If a COLLISION occurs, all active nodes with $b_1 = 0$ send RTB packets in the next mini-slot. The general rule on the $(i+2)$th mini-slot, $1 \le i \le M-1$, is that all active nodes with $b_i = 0$ send RTBs; at the end of the mini-slot, if a COLLISION is alarmed, all active nodes with $b_i = 1$ are backlogged and wait for the next slot; while a NULL is detected, all active nodes with $b_i = 1$ remain active in the next mini-slot. This process continues running until a SINGLE is recognized. Clearly, at the end of the collision resolution process, only the active node with the lowest-numbered *temporary_ID* will be the winner. To ensure fairness, each node subtracts one (mod N) from its current *temporary_ID* at the end of every slot. The advantage of the deterministic approach is that a winner is guaranteed to be elected if at least one active node exists. However, only the partial fairness can be achieved because of the multihop characteristic in ad-hoc networks. Besides, the value of M by the deterministic approach may be larger than that by the randomized approach.

In packet transmission phase, every winner in every grid in the leader election phase starts to transmit. Since simultaneous reception of packets on other channels is not affected, all nodes can gain the data concurrently. The control packet length is typically smaller than the data packet length, it is worthwhile taking multiple mini-slots to compete for the access right. To sum up, our hybrid MAC protocol is similar to the leader election among active nodes within each gird in every slot.

3 Performance Simulations

Two bandwidth models have been proposed in [14] to evaluate the network throughput performance for multichannel ad-hoc networks. (i) *Fixed-channel-bandwidth*: Each channel has a fixed bandwidth. The more the channels, the more bandwidth the network can potentially use. This model is especially suitable for CDMA environments. (ii) *Fixed-total-bandwidth*: The total bandwidth offered to the network is fixed. With more channels, each channel will have less bandwidth. This model is especially suitable for FDMA environments.

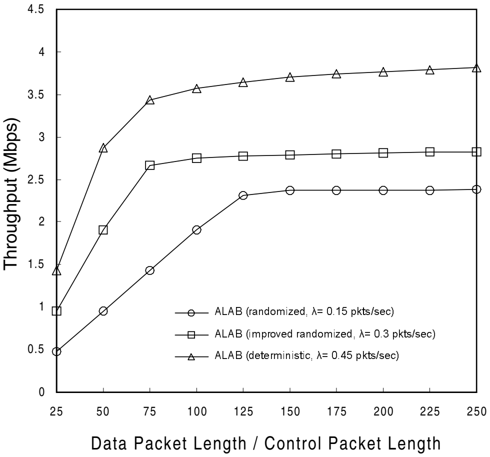


Fig. 3. L_d/L_c versus throughput under the fixed-total-bandwidth model. ($\eta = 8$ and $|\mathcal{G}| = 10 \times 10$.)

Due to space limitations, mathematical analysis can be referred to our technical report [6]. In this section, we report the simulation results. We use the *fixed-increment time advance* approach [10] for our discrete-event simulation model to evaluate the performance of ALAB. We have developed a simulator by C++. The ad-hoc network is simulated by placing N nodes randomly and uniformly within a bounded geographic region. The *geographic region size* ($|\mathcal{G}| = \frac{A}{d^2}$) is measured by the number of grids. The transmission range of all simulated nodes is r meters. The control packet length L_c including the guard time is 20 bytes and the data packet length L_d is a multiple of L_c. Network traffic was generated according to a Poisson arrival process with a mean of λ packets per second, and uniformly distributed among the nodes. If the fixed-channel-bandwidth model is assumed, each channel's bandwidth is 1 Mbps. If the fixed-total-bandwidth is assumed, the total bandwidth is 1 Mbps. We will consider the effect of node density on the performance instead of the average degree, where the *node density of the grid plane* ($\eta = \frac{N}{|\mathcal{G}|}$) is defined as the average number of nodes per grid.

A) Effect of Data Packet Length: In Fig. 3, we show the effect of the ratio L_d/L_c on the throughput performance under the fixed-total-bandwidth model. In this experiment, we fix η and $\mathcal{G}$ as 8 and 10×10, respectively. We can see that when $L_d/L_c \leq 125$, the throughput is highly promoted with the increasing length of data packet. This is because each successful leader election process can schedule more data bits to be sent. However, if we further increase the ratio L_d/L_c, the throughput of ALAB will be saturated at a certain point. As shown in Fig. 3, as both offered load and L_d/L_c increase, the throughput of ALAB (deterministic collision resolution approach) approaches the network capacity.

B) Effect of Arrival Rate and Bandwidth Models: In this experiment, we assume that $N = 512$, $r = 2d$, $|\mathcal{G}| = 8 \times 8$, $\eta = 8$, and $L_d/L_c = 50$. Fig. 4 and

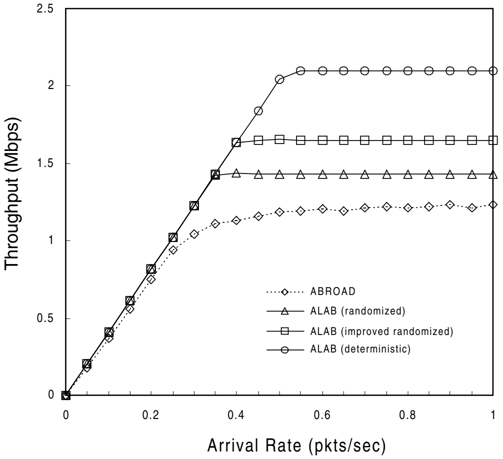


Fig. 4. Arrival rate versus throughput under the fixed-total-bandwidth model. ($N = 512$, $r = 2d$, $\eta = 8$, $|\mathcal{G}| = 8 \times 8$, and $L_d/L_c = 50$.)

5 show the throughput versus the offered load under the fixed-total-bandwidth model and under the fixed-channel-bandwidth model respectively. Especially, even under the fixed-total-bandwidth model, we find a 70% increase in the peak performance for ALAB over ABROAD, which delivers superior performance than TDMA, IEEE 802.11, and ADAPT [4,5]. The reasons are three-fold. (i) In ALAB, via the location-aware channel assignment scheme, the number of potential interfering terminals is significantly reduced from the size of *two-hop neighborhood* to the size of *intra-grid neighborhood.* (ii) Via the leader election process in ALAB, the probability for a node to reserve a slot is highly boosted. (iii) In such a crowed environment, the *erasure effect* [8] or deadlocks also cause the performance of ABROAD degradation. However, it is not very fair to compare ABROAD and ALAB because of their different assumptions on the transceivers. In Fig. 5, we see that the ALAB protocol with the deterministic collision resolution approach performs best since an active node is guaranteed to be elected (if it exists) in a grid in a slot.

C) Effect of Node Density: Fig. 6 shows the throughput versus node density and arrival rate under the fixed-channel-bandwidth model. We use $N = 256$ and $L_d/L_c = 75$. We see that as the node density decreases and/or the traffic load increases, the throughput increases monotonically and is finally saturated at a certain point. Especially, we find that when $\lambda = 15 \sim 20$ and $\eta = 4 \sim 16$, the deterministic collision resolution approach yields about 27.67% $\sim$ 56.67% improvement in the throughput, as compared with the randomized one. This is reasonable due to the uncertainty in the leader election phase by the randomized approach. Given fixed $\mathcal{A}$ and N, decreasing the node density will promote the throughput; meanwhile, it will cause the number of grids increase. Since we restrict $\sqrt{2}d \le r \le 2d$ in our design, a larger number of grids implies a shorter

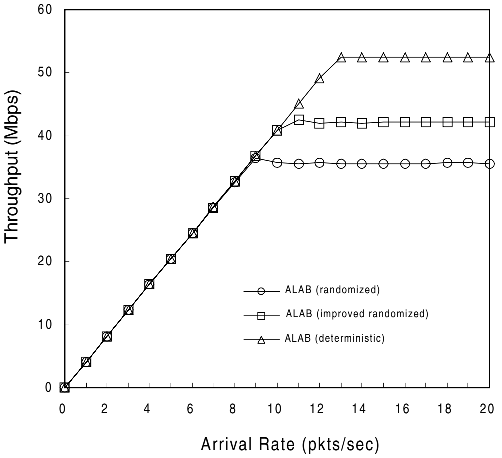


Fig. 5. Arrival rate versus throughput under the fixed-channel-bandwidth model. ($N = 512$, $\eta = 8$, $|\mathcal{G}| = 8 \times 8$, and $L_d/L_c = 50$.)

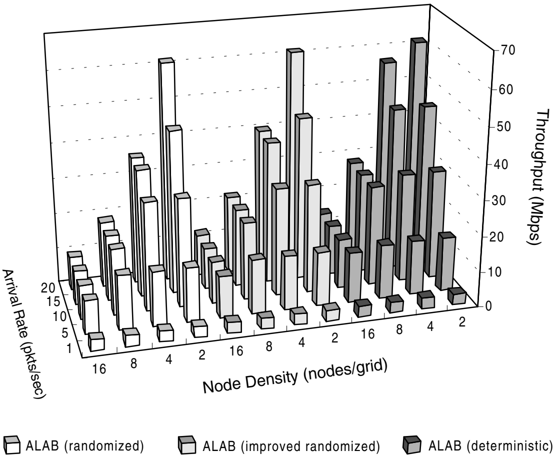


Fig. 6. Throughput versus node density and arrival rate under the fixed-channel-bandwidth model. ($N = 256$ and $L_d/L_c = 75$.)

transmission range. From the perspective of the routing performance, this will result in more hops from sources to destinations. To sum up, determining the optimal values of r and d is not an easy task.

D) Effect of Node ID Distribution: In all the above experiments, we have observed that the ALAB protocol with the deterministic collision resolution approach performs best. However, its collision resolution method highly depends on the distribution of the node IDs. In spite of the multihop characteristic in ad-hoc networks, each contending station should receive an equal share of the transmission bandwidth. We conduct an experiment to understand this fairness issue. We use $N = 16$, $\eta = 4$, $|\mathcal{G}| = 4$, and $L_d/L_c = 75$. Four sample nodes intended for our observation are 0000, 0001, 1010, and 1011. Furthermore, we assume that they are located in a same grid. Fig. 7 shows the simulation result under the fixed-channel-bandwidth model. We see that as the offered load increases, the performance range of the sample nodes increases significantly. That is, the unfairness problem becomes serious when traffic load is heavy. Therefore, if fairness is critical, the ALAB protocol with the improved randomized collision resolution approach may be a compromise solution.

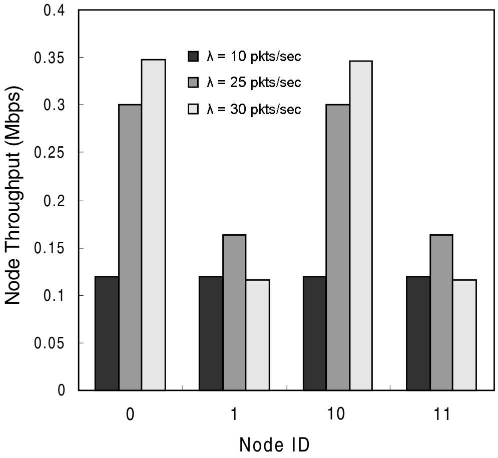


Fig. 7. Node ID versus node throughput under the fixed-channel-bandwidth model. ($N = 16$, $\eta = 4$, $|\mathcal{G}| = 4$, and $L_d/L_c = 75$.)

4 Conclusions

In this paper, we have proposed a new adaptive location-aware MAC protocol, called ALAB, for link-level broadcast support in multichannel MANETs. By virtue of GPS and channel assign scheme, all nodes within the same grid are able to maintain a consistent channel view. Due to the channel consistency in every grid, no deadlock or hidden terminal problems will exist. ALAB is scalable and mobility-transparent since it does not require any link state information. Using the ternary channel feedback information, our novel hybrid broadcast scheme can achieve high throughput performance. In principle, ALAB tries to combine both of the advantages of the allocation- and contention-based protocols and overcomes their individual drawbacks. ALAB has deterministic access guarantees by its base TDMA allocation protocol while providing flexible and efficient bandwidth management by reclaiming unused slots through the stable tree-splitting algorithms. Extensive experimental results have been conducted, which take many factors, such as channel bandwidth models, arrival rate, data packet length, node density, and fairness, into consideration. Both analysis [6] and simulation results do confirm the advantage of our scheme over other MAC protocols, such as IEEE 802.11, ADAPT [4], and ABROAD [5], even under the fixed-total-bandwidth model. All these results make ALAB a promising protocol to enhance the performance of the MANET.

References

1. G. Anastasi, L. Lenzini, and E. Mingozzi. HIPERLAN/1 MAC protocol: Stability and performance analysis. *IEEE Journal on Selected Areas in Communications*, Vol. 18, No. 9, Sep., (2000) 1787–1798.
2. D. Bertsekas and R. Gallager. *Data Networks, Second Edition*, Prentice-Hall, 1992.
3. I. Chlamtac and A. Faragó. An optimal channel access protocol with multiple reception capacity. *IEEE Trans. on Computers*, Vol. 43, No. 4, (1994) 480–484.
4. I. Chlamtac, A. Faragó, A. D. Myers, V. R. Syrotiuk, and G. Záruba. ADAPT: a dynamically self-adjusting media access control protocol for ad hoc networks. *GLOBECOM '99*, Vol. 1A, (1999) 11–15.
5. I. Chlamtac, A. D. Myers, V. R. Syrotiuk, and G. Záruba. An adaptive medium access control (MAC) protocol for reliable broadcast in wireless networks. *IEEE International Conference on Communications*, Vol. 3, (2000) 1692–1696.
6. Z.-T. Chou, C.-C. Hsu, and F.-C. Lin. An adaptive location-aware MAC Protocol for multichannel multihop ad-hoc networks. *Technical Report*, National Taiwan University, 2001.
7. R. Garcés and J.J. Garcia-Luna-Aceves. Collision avoidance and resolution multiple access with transmission queues. *Wireless Networks*, Vol. 5, (1999) 95–109.
8. T.-C. Hou and T.-J. Tsai. An access-based clustering protocol for multihop wireless ad hoc networks. *IEEE Journal on Selected Areas in Communications*, Vol. 19, No. 7, July, (2001) 1201–1210.
9. J.-H. Ju and V. O. K. Li. TDMA scheduling design of multihop packet radio networks based on Latin squares. *IEEE Journal on Selected Areas in Communications*, Vol. 7, No. 8, Aug., (1999) 1345–1352.
10. Averill M. Law and W. David Kelton. *Simulation Modeling and Analysis, Third Edition*, McGrraw-Hill Book Company Inc., 2000.
11. M. K. Marina, G. D. Kondylis, and U. C. Kozat. RBRP: A robust broadcast reservation protocol for mobile ad hoc networks. *IEEE International Conference on Communications*, Vol. 3, (2001) 878–885.
12. K. Nakano and S. Olariu. Randomized initialization protocols for ad hoc networks. *IEEE Trans. Parallel and Distributed Systems*, Vol. 11, No. 7, July, (2000) 749–759.
13. Z. Tang and J.J. Garcia-Luna-Aceves. A protocol for topology-dependent transmission scheduling in wireless networks. *IEEE Wireless Communications and Networking Conference*, Vol. 3, (1999) 1333–1337.
14. Y.-C. Tseng, S.-L. Wu, C.-M. Chao, and J.-P. Sheu. Location-aware channel assignment for a multi-channel mobile ad hoc network. *ICS2000 Workshop on Computer Networks, Internet, and Multimedia*, 2000.
15. C. Zhu and M. Corson. A five-phase reservation protocol (FPRP) for mobile ad hoc networks. *Wireless Networks*, Vol. 7, (2001) 371–384.

Capacity Assignment in Bluetooth Scatternets – Analysis and Algorithms

Gil Zussman and Adrian Segall

Department of Electrical Engineering
Technion – Israel Institute of Technology, Haifa 32000, Israel
{gilz@tx, segall@ee}.technion.ac.il
http://www.comnet.technion.ac.il/segall

Abstract. Bluetooth enables portable electronic devices to communicate wirelessly via short-range ad-hoc networks. Initially Bluetooth will be used as a replacement for point-to-(multi)point cables. However, in due course, there will be a need for forming multihop ad-hoc networks over Bluetooth, referred to as scatternets. This paper investigates the capacity assignment problem in Bluetooth scatternets. The problem arises primarily from the special characteristics of the network and its solution requires new protocols. We formulate it as a problem of minimizing a convex function over a polytope contained in the matching polytope. Then, we develop an optimal algorithm which is similar to the well-known flow deviation algorithm and that calls for solving a maximum-weight matching problem at each iteration. Finally, a heuristic algorithm with a relatively low complexity is developed.

Keywords: Bluetooth, Scatternet, Capacity assignment, Capacity allocation, Scheduling, Personal Area Networks (PAN)

1 Introduction

Recently, much attention has been given to the research and development of Personal Area Networks (PAN). These networks are comprised of personal devices, such as cellular phones, PDAs and laptops, in close proximity to each other. Bluetooth is an emerging PAN technology which enables portable devices to connect and communicate wirelessly via short-range ad-hoc networks [5],[6],[11]. Since its announcement in 1998, the Bluetooth technology has attracted a vast amount of research. However, the issue of capacity assignment in Bluetooth networks has been rarely investigated. Moreover, most of the research regarding network protocols has been done via simulation. In this paper we formulate an analytical model for the analysis of the capacity assignment problem and propose optimal and heuristic algorithms for its solution.

Bluetooth utilizes a short-range radio link. Since the radio link is based on frequency-hop spread spectrum, multiple channels (frequency hopping sequences) can co-exist in the same wide band without interfering with each other. Two or more units sharing the same channel form a *piconet*, where one unit acts as a *master* controlling the communication in the piconet and the others act as *slaves*.

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 411-422, 2002.

Bluetooth channels use a frequency-hop/time-division-duplex (FH/TDD) scheme. The channel is divided into 625-μsec intervals called *slots*. The master-to-slave transmission starts in even-numbered slots, while the slave-to-master transmission starts in odd-numbered slots. Masters and slaves are allowed to send 1,3 or 5 slots *packets* which are transmitted in consecutive slots. A slave is allowed to start transmission in a given slot if the master has addressed it in the preceding slot. Information can only be exchanged between a master and a slave, i.e. there is no direct communication between slaves. Although packets can carry synchronous information (voice link) or asynchronous information (data link), in this paper we concentrate on networks in which only data links are used.

Multiple piconets in the same area form a *scatternet*. Since Bluetooth uses packet-based communications over slotted links, it is possible to interconnect different piconets in the same scatternet. Hence, a unit can participate in different piconets, on a time-sharing basis, and even change its role when moving from one piconet to another. We will refer to such a unit as a *bridge*. For example, a bridge can be a master in one piconet and a slave in another piconet. However, a unit cannot be a master in more than one piconet.

Initially Bluetooth piconets will be used as a replacement for point-to-(multi)point cables. However, in due course, there will be a need for multihop ad-hoc networks (scatternets). Due to the special characteristics of such networks, many theoretical and practical questions regarding the scatternet performance are raised. Nevertheless, only a few aspects of the scatternet performance have been studied. Two issues that received relatively much attention are: research regarding scatternet topology and development of efficient scatternet formation protocols (e.g. [4],[13]).

Much attention has also been given to scheduling algorithms for piconets and scatternets. In the Bluetooth specifications [5], the capacity allocation by the master to each link in its piconet is left open. The master schedules the traffic within a *piconet* by means of polling and determines how bandwidth capacity is to be distributed among the slaves. Numerous heuristic scheduling algorithms for piconets have been proposed and evaluated via simulation (e.g. [7],[8]). In [11] an overall architecture for handling scheduling in a *scatternet* has been presented and a family of inter-piconet scheduling algorithms (algorithms for masters and bridges) has been introduced. Inter-piconet scheduling algorithms have also been proposed in [1] and [16].

Although scatternet formation as well as piconet and scatternet scheduling have been studied, the issue of *capacity assignment* in Bluetooth scatternets has not been investigated. Moreover, Baatz et al. [1] who made an attempt to deal with it have indicated that it is a complex issue.[1] Capacity assignment in communication networks focuses on finding the best possible set of link capacities that satisfies the traffic requirements while minimizing some performance measure (such as average delay). We envision that in the future, capacity assignment protocols will start operating once the scatternet is formed and will determine link capacities that will be dynamically allocated by scheduling protocols. Thus, *capacity assignment protocols* are the

[1] In [1] the term *piconet presence schedule* is used to refer to a notion similar to *capacity assignment*.

missing link between scatternet formation and scatternet scheduling protocols. A correct use of such protocols will improve the utilization of the scatternet bandwidth. We also anticipate that the optimal solution of the *capacity assignment problem* will improve the evaluation of heuristic scatternet scheduling algorithms.

Most models of capacity assignment in communication networks deal mainly with static networks in which a cost is associated with each level of link capacity (see [3] for a review of models). The following discussion shows that there is a need to study the capacity assignment problem in Bluetooth scatternets in a different manner:

- In contrast with a wired and static network, in an ad-hoc network, there is no central authority responsible for network optimization and there is no monetary cost associated with each level of link capacity.
- The nature of the network allows frequent changes in the topology and requires frequent changes in the capacities assigned to every link.
- There are constraints imposed by the tight master-slave coupling and by the time-division-duplex (TDD) scheme.
- Unlike other ad-hoc networks technologies in which all nodes within direct communication from each other share a common channel, in Bluetooth only a subgroup of nodes (piconet) shares a common channel and capacity has to be allocated to each link.

A scatternet capacity assignment protocol has to determine the capacities that each master should allocate in its own piconet, such that the network performance will be optimized. Currently, our major interest is in algorithms for quasi-static capacity assignment that will minimize the average delay in the scatternet. The analysis is based on a static model with stationary flows and unchanging topology. To the best of our knowledge, the work presented in this paper is the first attempt to analytically study the capacity assignment problem in Bluetooth scatternets.

In this paper we focus on formulating the problem and developing centralized algorithms. The development of the distributed protocols is subject of further research.

In the sequel we formulate the scatternet capacity assignment problem as a minimization of a convex function over a polytope contained in the polytope of the well-known *matching problem* [14, p. 608] and show that different formulations apply to bipartite and nonbipartite scatternets. The methodology used by Gerla et al. [9],[15] is used in order to develop an *optimal scatternet capacity assignment algorithm* which is similar to the *flow deviation algorithm* [3, p. 458]. The main difference between the algorithms is that at each iteration there is a need to solve a *maximum-weight matching problem* instead of a *shortest path problem*. Finally, we introduce a *heuristic algorithm* whose complexity is much lower than the complexity of the optimal algorithm and whose performance is often close to that of the optimal algorithm. Due to space constraints, numerical results are not presented in this paper and the proofs are omitted. Yet, numerical examples and the proofs can be found in [18].

This paper is organized as follows. In Section 2, we present the model and in Section 3 we formulate the scatternet capacity assignment problem for bipartite and nonbipartite scatternets. An algorithm for obtaining the optimal solution of the problem is presented in Section 4. In Section 5, we develop a heuristic algorithm for bipartite scatternets and in Section 6 we summarize the main results.

2 Model and Preliminaries

Consider the connected undirected scatternet graph $G = (N,L)$. N will denote the collection of *nodes* $\{1,2,\ldots,n\}$. Each of the nodes could be a master, a slave, or a bridge. The *bi-directional link* connecting nodes i and j will be denoted by (i,j) and the collection of bi-directional links will be denoted by L. For each node i, denote by $Z(i)$ the collection of its neighbors. We denote by $L(U)$ ($U \subseteq N$) the collection of links connecting nodes in U.

Usually, capacity assignment protocols deal with the allocation of capacity to directional links. However, due to the tight coupling of the uplink and downlink in Bluetooth piconets[2], we concentrate on the total bi-directional link capacity. Hence, we assume that the average packet delay on a link is a function of the total link flow and the total link capacity. An equivalent assumption is that the uplink and the downlink flows are equal (symmetrical flows).

Let F_{ij} be the average bi-directional flow on link (i,j) and let C_{ij} be the capacity of link (i,j) (the units of F and C are bits/second). We assume that at every link the average bi-directional flow is positive ($F_{ij} > 0 \;\; \forall (i,j) \in L$). We define f_{ij} as the ratio between F_{ij} and the maximal possible flow on a Bluetooth link when using a given type of packets[3]. We also define c_{ij} as the ratio between C_{ij} and the maximal possible capacity of a link. It is obvious that $0 < f_{ij} \le 1$ and that $0 \le c_{ij} \le 1$. In the sequel, f_{ij} will be referred to as the *flow on link* (i,j) and c_{ij} will be referred to as the *capacity of link* (i,j). Accordingly, $\bar{c}$ will denote the vector of the link capacities and will be referred to as the *capacity vector*.

The objective of the capacity assignment algorithms, described in this paper, is to minimize the average delay in the scatternet. We define D_{ij} as the total delay per unit time of all traffic passing through link (i,j), namely:

Definition 1. *D_{ij} is the average delay per unit of the traffic multiplied by the amount of traffic per unit time transmitted over link (i,j).*

We assume that D_{ij} is a function of the link capacity c_{ij} only. We should point out that the optimal algorithm requires no explicit knowledge of the function $D_{ij}(c_{ij})$. We shall need to assume only the following reasonable properties of the function $D_{ij}(\cdot)$.

Definition 2. *$D_{ij}(\cdot)$ is defined such that all the following holds:*

1. *D_{ij} is a nonnegative continuous decreasing function of c_{ij} with continuous first and second derivatives.*
2. *D_{ij} is convex.*
3. $\lim\limits_{c_{ij} \to f_{ij}} D_{ij}(c_{ij}) = \infty$
4. $D_{ij}'(c_{ij}) < 0$ *for all c_{ij} where D_{ij}' is the derivative of D_{ij}.*

[2] A slave is allowed to start transmission only after a master addressed it in the preceding slot.

[3] For example, currently the maximal flow on a symmetrical link, when using five slots unprotected data packets (DH5), is 867.8 Kbits/second.

Using Definition 1, we shall now define the total delay in the network.

Definition 3. *The* total delay in the network per unit time *is denoted by* D_T *and is given by*:

$$D_T = \sum_{(i,j)\in L} D_{ij}(c_{ij})$$

Since the total traffic in the network is independent of the capacity assignment procedure, we can minimize the average delay in the network by minimizing D_T. A capacity vector that achieves the minimal average delay will be denoted by $\overline{c}^*$.

A *capacity assignment algorithm* has to determine what portion of the slots should be allocated to each master-slave link. On the other hand, a *scheduling algorithm* has to determine which master-slave links should use any given slot pair. Hence, we define a scheduling algorithm as follows.

Definition 4. *A* Scheduling Algorithm *determines how each slot pair is allocated. It does not allow transmission on two adjacent links in the same slot pair.*

The Bluetooth specifications [5] do not require that different masters' clocks will be synchronized. Since the clocks are not synchronized a guard time is needed in the process of moving a bridge from one piconet to another. Yet, in order to formulate a simple analytical model we assume that the *guard times are negligible*. This assumption allows us to consider a scheduling algorithm for the whole scatternet.

3 Formulation of the Problem

Scatternet graphs can be *bipartite graphs* or *nonbipartite graphs* [4] (a graph is called bipartite if there is a partition of the nodes into two disjoint sets S and T such that each edge joins a node in S to a node in T [14, p. 50]). Any scatternet graph in which no master is allowed to be a bridge is necessarily bipartite. For example, the scatternet graph described in Fig. 1-A is bipartite. Even if a master is allowed to be a bridge, the scatternet may be bipartite (e.g. Fig. 1-B). Obviously, if a master is allowed to be a bridge, the scatternet graph may be nonbipartite (e.g. Fig 1-C).

In this section, we shall formulate the *capacity assignment problem* for bipartite and nonbipartite scatternets. We will show that the formulation for nonbipartite scatternets is more complex than the formulation for bipartite scatternets.

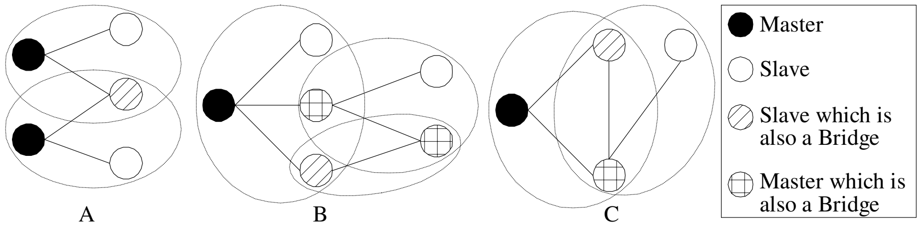


Fig. 1. Scatternet graphs – A bipartite scatternet in which no master is also a bridge (A), a bipartite scatternet in which a master is also a bridge (B), and a nonbipartite scatternet (C)

3.1 Bipartite Scatternets

When a bipartite scatternet graph is given, the nodes can be partitioned into two sets S and T such that no two nodes in S or in T are adjacent. Accordingly, the problem of *scatternet capacity assignment in bipartite graphs* (SCAB) is formulated as follows.

Problem SCAB
Given: Topology of a bipartite graph and flows (f_{ij}).
Objective: Find capacities (c_{ij}) such that the average packet delay is minimized:

$$\min D_T = \min \sum_{(i,j)\in L} D_{ij}(c_{ij}) \tag{1}$$

Subject to:

$$c_{ij} > f_{ij} \quad \forall (i,j) \in L \tag{2}$$

$$\sum_{j\in Z(i)} c_{ij} \le 1 \quad \forall i \in S \tag{3}$$

$$\sum_{j\in Z(i)} c_{ij} \le 1 \quad \forall i \in T \tag{4}$$

The first set of constraints (2) is obvious. Constraints (3) and (4) result from the TDD scheme and reflect the fact that the total capacity of the links connected to a node cannot exceed the maximal capacity of a link. Due to the assumption that the *guard times are negligible*, in (3) and (4) we neglect the time needed in the process of moving a bridge from one piconet to another. Notice that it is easy to see that the polytope defined by (2) - (4) is contained in the *bipartite matching* polytope [14].

3.2 Nonbipartite Scatternets

We shall now show that a formulation similar to the formulation of Problem SCAB is not valid for nonbipartite scatternets. A simple example of a nonbipartite scatternet, given in [1], is illustrated in Fig. 2-A. Constraint (2) and the constraint:

$$\sum_{j\in Z(i)} c_{ij} \le 1 \quad \forall i \in N \tag{5}$$

are not sufficient in order for the capacity vector to be feasible in this example. The capacities described in Fig. 2-A satisfy (2) and (5) but are not feasible because in any scheduling algorithm no two neighboring links can be used simultaneously. If links (1,2) and (1,3) are in use for distinct halves of the available time slots, there are no free slots in which link (2,3) can be in use. Thus, if $c_{12} = 0.5$ and $c_{13} = 0.5$, there is no feasible way to assign any capacity to link (2,3), i.e., there is no scheduling algorithm that can allocate the capacities described in the figure.

Baatz et al. [1] suggest that a methodology for finding a *feasible* (not necessarily efficient) capacity assignment[4] will be based on minimum coloring of a graph. They do not develop this methodology and indicate that: "*the example gives an idea of how*

[4] Baatz et al. [1] refer to *piconet presence schedule* instead of *capacity assignment.* A piconet presence schedule determines in which parts of its' time a node is present in each piconet. It is very similar to link capacity assignment as it is described in this paper.

complex the determination of piconet presence schedules may get". We propose a formulation of the problem that is based on the formulations of Problem SCAB and the *matching problem* [14], and that allows obtaining an optimal capacity allocation.

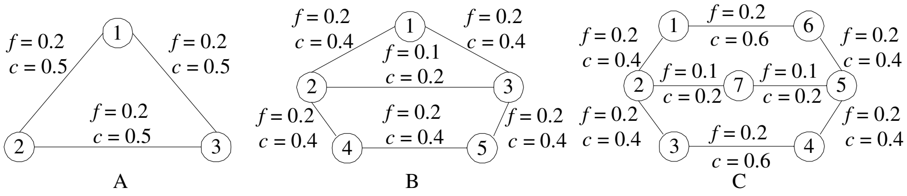


Fig. 2. Examples of scatternets with capacity vectors which are not feasible

It is now obvious that the formulation of the capacity assignment problem for non-bipartite scatternets requires additional constraints to the constraints described in Problem SCAB. For example, one could conclude that the capacity of the links composing the cycle described in Fig. 2-A should not exceed 1. Moreover, one could further conclude that the total capacity of links composing any odd cycle should not exceed: (|links|-1)/2. Namely:

$$\sum_{(i,j)\in C} c_{ij} \le \left(|C|-1\right)/2 \quad \forall C \subseteq L,\ C \text{ odd cycle} \qquad (6)$$

However, in the examples given in Fig. 2-B and Fig. 2-C, although the capacities satisfy (6), they cannot be scheduled in any way. Thus, in the following theorem we define a new set of constraints such that the capacity of links connecting nodes in any odd set of nodes U will not exceed $(|U|-1)/2$.[5] These constraints and the proof of the theorem are based on the properties of the matching problem [10],[14].

Theorem 1. *The capacity vector must satisfy (2),(5), and the following constraints:*

$$\sum_{(i,j)\in L(U)} c_{ij} \le \left(|U|-1\right)/2 \quad \forall U \subseteq N,\ |U|\,odd,\ |U| \ge 3 \qquad (7)$$

The proof appears in [18].

The *scatternet capacity assignment* problem (SCA) can now be formulated as follows (for bipartite graphs it reduces to Problem SCAB).

Problem SCA
Given: Topology and flows (f_{ij}).
Objective: Find capacities (c_{ij}) such that the average packet delay is minimized: (1)
Subject to: (2),(5) and (7)

The constraints (2),(5) and (7) form a *convex set* which is included in the matching polytope corresponding to the scatternet graph (for bipartite scatternets these constraints reduce to constraints (2) - (4) described in Problem SCAB.). This set consists of all the *feasible capacity vectors* ($\overline{c}$). Up to now we have not shown that a

[5] We note that a similar observation has been recently independently made by Tassiulas and Sarkar [17] who have considered the problem of max-min fair scheduling in scatternets.

feasible capacity vector has a corresponding scheduling algorithm. Namely, that it is possible to determine which links are used in each slot pair such that no two adjacent links are active at the same slot pair and the capacity used by each link is as defined by the capacity vector ($\overline{c}$). This result is shown by the following proposition. We note that the proof of the proposition and the transformation of a capacity vector to a scheduling algorithm are based on the fact that the vertices of the matching polytope are composed of (0,1) variables and on an algorithm described in [10].

Proposition 1. *If a capacity vector $\overline{c}$ satisfies (2),(5) and (7), there is a corresponding scheduling algorithm.*
The proof appears in [18].

4 Optimal Algorithm for Problems SCA and SCAB

In this section a *centralized scatternet capacity assignment algorithm* for finding an optimal solution of Problem SCA, defined in Section 3.2, is introduced.[6] The algorithm is based on the conditional gradient method also known as the Frank-Wolfe method [2, p. 215], which was used for the development of the flow deviation algorithm [3, p. 458]. Gerla et al. [9],[15] have used the Frank-Wolfe method in order to develop bandwidth allocation algorithms for ATM networks. Following their approach, we shall now describe the optimality conditions and the algorithm.

Since the objective of Problem SCA is to minimize a convex function (D_T) over a convex set ((2),(5) and (7)), any local minimum is a global minimum. Thus, necessary and sufficient conditions for the capacity vector $\overline{c}^*$ to be a global minimum are formulated as follows (the following proposition is derived from a well-known theorem [2, p. 194] and, therefore, its proof is omitted).

Proposition 2. *The capacity vector $\overline{c}^*$ minimizes the average delay for Problem SCA, if and only if:*

- *$\overline{c}^*$ satisfies constraints (2),(5) and (7) of Problem SCA.*
- *There are no feasible directions of descent at $\overline{c}^*$; i.e. there does not exist $\overline{c}$ such that*[7]:

$$\nabla D_T(\overline{c}^*)(\overline{c}-\overline{c}^*)<0 \tag{8}$$

$$\sum_{j\in Z(i)} c_{ij} \le 1 \quad \forall i\in N \tag{9}$$

$$\sum_{(i,j)\in L(U)} c_{ij} \le \left(|U|-1\right)/2 \quad \forall U\subseteq N,\ |U|\,odd,\ |U|\ge 3 \tag{10}$$

Proposition 2 suggests a steepest descent algorithm in which we can find a feasible direction of descent $\overline{c}$ at any feasible point $\overline{c}^K$ by solving the problem:

$$\min \nabla D_T(\overline{c}^K)\overline{c} \tag{11}$$

subject to - (9),(10) and:

[6] The algorithm for the solution of Problem SCAB is similar (the changes are outlined below).

[7] $\nabla D_T(\overline{c}^*)$ is the gradient of D_T with respect to $\overline{c}$ evaluated at $\overline{c}^*$.

$$c_{ij} \geq 0 \quad \forall (i,j) \in L \tag{12}$$

Since the constraint set (10) may include exponentially many constraints, this problem cannot be easily solved using a linear programming algorithm. Yet, since $D_{ij}'(c_{ij}) < 0$ for all c_{ij} (according to Definition 2.4), the formulation of the problem conforms to the formulation of the *maximum-weight matching* problem [14, p. 610], which has a polynomial-time algorithm ($O(n^3)$):

$$\max\left[-\nabla D_T(\overline{c}^K)\overline{c}\right] \tag{13}$$

subject to:

$$\sum_{j\in Z(i)} c_{ij} \leq 1 \quad \forall i \in N \tag{14}$$

$$c_{ij} \in \{0,1\} \quad \forall (i,j) \in L \tag{15}$$

This result and Proposition 2 are the basis for the optimal algorithm, described in Fig 3. The input to the algorithm is the topology, the flows (f_{ij}), a feasible initial solution ($\overline{c}^0$), and the tolerance (t). The output is the optimal capacity vector: $\overline{c}^*$.

```
1   Set $K = 0$
2   Find the vector $\overline{c}^\#$ - the optimal solution of (13) - (15) (i.e. solve a maximum-weight matching problem)
3   Find the value $\alpha^*$ that minimizes $D_T(\alpha\overline{c}^K + (1-\alpha)\overline{c}^\#)$ ($\alpha^*$ may be obtained by any line search method [2, p. 723])
4   Set $\overline{c}^{K+1} = \alpha^*\overline{c}^K + (1-\alpha^*)\overline{c}^\#$
5   If $\nabla D_T(\overline{c}^K)(\overline{c}^K - \overline{c}^\#) \leq t$ then stop
6   Else set $K = K+1$ and go to 2
```

Fig. 3. An algorithm for obtaining the optimal solution to Problem SCA

We emphasize that unlike the flow deviation algorithm, in which at each iteration a feasible direction is found by solving a shortest path problem, in the capacity assignment algorithm there is a need to solve a maximum-weight matching problem at each iteration. In case the algorithm is applied to Problem SCAB, there is a need to solve a *bipartite* maximum-weight matching problem.

5 Heuristic Algorithm for Problem SCAB

When considering *bipartite scatternets* (Problem SCAB), the initial solution for the optimal algorithm can be obtained using a low complexity *heuristic centralized scatternet capacity assignment* algorithm, presented in this section. In our experiments (see [18]), the results of the heuristic algorithm are very close to the optimal results.

The algorithm is based on the assumption that the delay function conforms to *Kleinrock's independence approximation* [12], described in the following definition.

Definition 5. (Kleinrock's independence approximation) *When neglecting the propagation and processing delay, $D_{ij}(c_{ij})$ is given by*:

$$D_{ij}(c_{ij}) = \begin{cases} f_{ij}/(c_{ij} - f_{ij}) & c_{ij} > f_{ij} \\ \infty & c_{ij} \le f_{ij} \end{cases}$$

The algorithm assigns capacity to links connected to bridges and to masters which have at least two slaves. Accordingly, we define N' as follows:

Definition 6. *N' is a subgroup of N consisting of bridges and masters which have at least two slaves. Namely*: $N' = \{ i \mid i \in N \cap |j \in Z(i)| > 1 \}$.

We also define the *slack capacity* of a node as follows:

Definition 7. *The* slack capacity of node *i is the maximal capacity which can be added to links connected to the node. It is denoted by s_i and is given by*:

$$s_i = 1 - \sum_{j \in Z(i)} c_{ij}$$

Initially all the link capacities are equal to the flows on the links ($c_{ij} = f_{ij}$ $\forall (i,j) \in L$). The algorithm selects a node from the nodes in N' and allocates the slack capacity to some of the links connected to it. Then, it selects another node, allocates capacity and so on. Once a node (k) is selected, *the slack capacity of this node is allocated to its links whose capacities have not yet been assigned.* The slack capacity is assigned to these links according to the square root assignment [12, p. 20]:

$$c_{kj} = f_{kj} + \frac{s_k \sqrt{f_{kj}}}{\sum_{m:\, m \in Z(k), c_{km} = f_{km}} \sqrt{f_{km}}} \quad \forall j: j \in Z(k), c_{kj} = f_{kj} \tag{16}$$

There are various ways to define the process of *node selection*. For example, nodes can be selected according to their slack capacity or their average slack capacity. However, some of the possible selection methodologies require taking special measures in order to ensure that the obtained capacity vector is feasible (satisfies constraints (2) - (4) of Problem SCAB). We propose a simple selection methodology that guarantees a feasible capacity vector.

It can be shown that after capacity is assigned to a subgroup of the links connected to a node (i) (links whose capacities have not been assigned before), the delay derivatives ($D_{ij}'(c_{ij})$) of all these links will be equal. Accordingly, we define the *delay derivative of a node* as follows:

Definition 8. *The* delay derivative of node *i is proportional to the absolute values of the delay derivatives of the links connected to node i, whose capacities have not yet been assigned. Its value is computed as if node i has been selected as the node whose capacity has to be assigned and the capacities of these links have been assigned according to (16). It is denoted by d_i and is given by*:

$$d_i = \frac{\sum\limits_{m:\, m\in Z(i), c_{im}=f_{im}} \sqrt{f_{im}}}{s_i} \qquad (17)$$

Node k, whose link capacities are going to be assigned, is selected from the nodes in N' which are connected to links whose capacities have not yet been allocated. The delay derivatives (d_i's) of all these nodes are computed and the node with the largest derivative is selected. Thus, *the capacities of links with high absolute value of delay derivative, whose delay is more sensitive to the level of capacity, are assigned first.*

The algorithm, which is based on the above methodology, is described in Fig 4. The input is the topology and the flows (f_{ij}), and the output is a capacity vector: $\overline{c}$. It can be seen that the complexity of the algorithm is $O(n^2)$, which is about the complexity of an iteration in the optimal algorithm. Moreover, the following proposition shows that the capacity vector obtained by the algorithm is always feasible.

```
1   Set $c_{ij}=f_{ij} \quad \forall(i,j)\in L$
2   Set $k = \arg\max_{\substack{i\in N' \\ \exists m\in Z(i) \text{ such that } c_{im}=f_{im}}} \frac{\sum_{m:\, m\in Z(i), c_{im}=f_{im}} \sqrt{f_{im}}}{1-\sum_{m\in Z(i)} c_{im}}$
3   Set $c_{kj} = f_{kj} + \frac{s_k \sqrt{f_{kj}}}{\sum_{m:\, m\in Z(k), c_{km}=f_{km}} \sqrt{f_{km}}} \quad \forall j:\, j\in Z(k), c_{kj} = f_{kj}$
4   If there exists $(i,j)\in L$ such that $c_{ij}=f_{ij}$ then go to 2
5   Else stop
```

Fig. 4. An algorithm for obtaining a heuristic solution to Problem SCAB

Proposition 3. *The heuristic algorithm results in an allocation $\{\overline{c}\}$ that satisfies constraints (2) - (4) of Problem SCAB.*
The proof appears in [18].

6 Conclusions and Future Study

This paper presents an analytical study of the capacity assignment problem in Bluetooth scatternets. The problem has been formulated for bipartite and nonbipartite scatternets, using the properties of the matching polytope. Then, we have introduced a centralized algorithm for obtaining its optimal solution. A heuristic algorithm for the solution of the problem in bipartite scatternets, which has a relatively low complexity, has also been described. Several numerical examples can be found in [18].

The work presented here is the first approach towards an analysis of the scatternet performance. Hence, there are still many open problems to deal with. For example, *distributed protocols* are required for actual Bluetooth scatternets and, therefore, future study will focus on developing optimal and heuristic distributed protocols.

Moreover, in this paper we have made a few assumptions regarding the properties of the *delay function*. An analytical model for the computation of bounds on the delay is required in order to evaluate these assumptions. In addition, it might enable developing efficient piconet scheduling algorithms.

Finally, we note that a major future research direction is the development of capacity assignment protocols that will be able to deal with various quality of service requirements and to interact with scatternet formation, scheduling, and routing protocols.

References

1. Baatz, S., Frank, M., Kühl, C., Martini, P., Scholz, C.: Adaptive Scatternet Support for Bluetooth using Sniff Mode. Proc. IEEE LCN'01 (Nov. 2001)
2. Bertsekas, D.P.: Nonlinear Programming. Athena Scientific, Massachusetts (1999)
3. Bertsekas, D.P., Gallager, R.: Data Networks. Prentice Hall, New Jersey (1992)
4. Bhagwat, P., Rao, S.P.: On the Characterization of Bluetooth Scatternet Topologies. Submitted for publication (Feb. 2002)
5. Bluetooth Special Interest Group: Specification of the Bluetooth System - Version 1.1. (Feb. 2001)
6. Bray, J., Sturman, C.: Bluetooth connect without cables. Prentice Hall (2001)
7. Bruno, R., Conti, M., Gregori, E.: Wireless Access to Internet via Bluetooth: Performance Evaluation of the EDC Scheduling Algorithm. Proc. ACM WMI'01 (July 2001)
8. Das, A., Ghose, A., Razdan, A., Saran, H., Shorey, R.: Enhancing Performance of Asynchronous Data Traffic over the Bluetooth Wireless Ad-hoc Network. Proc. IEEE INFOCOM'01 (Apr. 2001)
9. Gerla, M., Monteiro, J.A.S., Pazos-Rangel, R.A.: Topology Design and Bandwidth Allocation in ATM Nets. IEEE JSAC **7** (Oct. 1989) 1253-1262
10. Hajek, B., Sasaki, G.: Link Scheduling in Polynomial Time. IEEE Trans. on Information Theory **34** (Sep. 1988) 910-917
11. Johansson, P., Kazantzidis, M., Kapoor, R., Gerla, M.: Bluetooth: An Enabler for Personal Area Networking. IEEE Network **15** (Sep./Oct. 2001) 28-37
12. Kleinrock, L.: Communication Nets: Stochastic Message Flow and Delay. McGraw-Hill, New York (1964)
13. Law, C., Mehta, A.M., Siu, K.Y.: Performance of a New Bluetooth Scatternet Formation Protocol. Proc. ACM MOBIHOC'01 (Oct. 2001)
14. Nemhauser, G.L., Wolsey, L.A.: Integer and Combinatorial Optimization. John Wiley and Sons (1988)
15. Pazos-Rangel, R.A., Gerla, M.: Express Pipe Networks. Proc. Global Telecommunications Conf. (1982) B2.3.1-5
16. Racz, A., Miklos, G., Kubinszky, F., Valko, A.: A Pseudo Random Coordinated Scheduling Algorithm for Bluetooth Scatternets. Proc. ACM MOBIHOC'01 (Oct. 2001)
17. Tassiulas, L., Sarkar, S.: Maxmin Fair Scheduling in Wireless Networks. Proc. IEEE INFOCOM'02 (to appear)
18. Zussman, G., Segall, A.: Capacity Assignment in Bluetooth Scatternets - Analysis and Algorithms. CCIT Report 355, Technion - Department of Electrical Engineering, (Oct. 2001) Available at URL: http://www.comnet.technion.ac.il/segall/Reports.html

Optimization-Based Congestion Control for Multicast Communications[⋆]

Jonathan K. Shapiro, Don Towsley, and Jim Kurose

Department of Computer Science
University of Massachusetts at Amherst

Abstract. Widespread deployment of multicast depends on the existence of congestion control protocols that are provably fair to unicast traffic. In this work, we present an optimization-based congestion control mechanism for single-rate multicast communication with provable fairness properties. The optimization-based approach attempts to find an allocation of rates that maximizes the aggregate utility of the network. We show that the utility of multicast sessions must be carefully defined if a widely accepted property of aggregate utility is to hold. Our definition of session utility amounts to maximizing a weighted sum of simple utility functions, with weights determined by the number of receivers. The fairness properties of the optimal rate allocation depend both on the weights and form of utility function used. We present analysis for idealized topologies showing that while our mechanism is not strictly fair to unicast, its unfairness can be controlled by appropriate choices of parameters.

1 Introduction

Widespread deployment of multicast communication in the Internet depends critically on the existence of practical congestion control mechanisms that allow multicast and unicast traffic to share network resources fairly. Most service providers recognize multicast as an essential service to support a range of emerging network applications including audio and video broadcasting, bulk data delivery, and teleconferencing. Nevertheless, these network operators have been reluctant to enable multicast delivery in their networks, often citing concerns about the congestion such traffic may introduce. There is a clear need for multicast congestion control algorithms that are provably fair to unicast traffic if these concerns are to be addressed. In this paper, we present a congestion control mechanism for single-rate multicast traffic based on an economic theory of resource allocation and show that although it is not strictly fair to unicast traffic, its unfairness is bounded and can be controlled.

We first formulate the multicast congestion control problem as a utility maximization problem, extending existing work for unicast. A naive, *sender oriented*,

[⋆] This work was supported in part by the National Science Foundation (NSF) under grant number ANI-9980552

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 423–442, 2002.

generalization of existing formulations for unicast treats single-source multicast sessions no differently from unicast sessions, modeling each by an unweighted utility function and maximizing the sum of session utilities. One problem with this naive approach is that it penalizes individual multicast sessions for using more network resources than unicast sessions without rewarding them for the bandwidth saved on links shared by multiple receivers. More serious than its unfairness to multicast sessions, the sender-oriented approach turns out to violate a generally accepted property of aggregate utility, namely, that the preference of the aggregate does not change if we simply measure utility on a different scale. This common-sense notion is why, for example, we reject as nonsense the statement that, as a group, residents of New York prefer a temperature of 70 degrees to 60 degrees Fahrenheit, but prefer a temperature of 15.5 to 21 degrees Celsius. If this invariance property is violated in the congestion control problem, the network will be controlled about an operating point determined by an arbitrary choice of utility scale. We introduce a *receiver-oriented* approach that uses session weights based on the number of receivers and preserves invariance under a change in utility scale. Moreover, we show that such an approach is necessary a neccessary condition for satisfying this property.

A consequence of adding session weights based on the number of receivers is that the resulting rate allocations tend to favor sessions with more receivers over those with fewer. Since the weighted sum does not remove the original penalty against sessions that use more resources, it is not immediately clear whether multicast sessions fare better or worse than unicast sessions under our formulation. We show that while our formulation favors multicast sessions, the resulting unfairness can be controlled and remains bounded in the simple network topologies we have considered.

Our work is based on a promising economics-inspired approach called *optimization based congestion control*, which casts the congestion problem as one of utility maximization (alternately, cost minimization). This approach provides an elegant theoretical framework in which congestion signals are interpreted as prices, network users are modeled as utility maximizers, and the network sets prices in such a way to drive a set of self-interested users toward an operating point at which their aggregate utility is maximized. Specific link service disciplines and rate-control algorithms at end-hosts can be thought of as components of a distributed computation to solve the global optimization problem. Thus, improvements in congestion control can proceed in a principled fashion, driven by improvements in the underlying optimization algorithm. While the optimization-based approach has received much attention [1,2,3,4,5,6,7,8], it has only recently been applied to multicast congestion control [9,10]. Many existing mulicast congestion control schemes [11,12] rely on heuristic techniques, such as adapting to a single receiver or a small group of representatives. In contrast, the optimization-based approach offers a formal foundation with which to develop congestion control mechanisms and understand their fairness properties and impact on the global behavior of the network.

The rest of this paper is structured as follows: In Section 2 we extend a unicast congestion control problem formulation to single-rate multicast. In Section 3 we consider multicast session utility functions in detail, presenting a axiomatic argument in favor of a particular definition. The fairness properties of our definition are analyzed in Sections 4-6 where we show that multicast sessions are favored over unicast sessions and present evidence that such unfairness can be controlled. We conclude by briefly discussing the development of practical control mechanisms based on our results and highlighting future work.

2 Problem Formulation

Optimization-based congestion control casts the problem of bandwidth sharing as one of utility maximization. Consider a network modeled as a set of directed links L, with capacity c_l for each link $l \in L$. Let $C = (c_l, l \in L)$. The workload for the network is generated by a set of sessions[1] S, which consume bandwidth. The set of links used by a particular session, s, is $L(s) \subseteq L$. The set of sessions using any particular link, l, is $S(l) \subseteq S$.

Each session is characterized by a utility function U_s, which is assumed to be increasing and concave in the session rate x_s. Session utility may also be a function of other parameters in addition to rate, such as number of receivers, but we will sometimes suppress these additional dependencies in the notation, writing $U_s(x_s)$. The network's objective is to optimize social welfare, defined as the sum of session utilities.

$$\max_{x_s \geq 0} \quad \sum_{s \in S} U_s(x_s) \tag{1}$$

$$\text{subject to} \sum_{s \in S(l)} x_s \leq c_l \quad l \in L \tag{2}$$

The problem (1-2) can be solved using convex optimization techniques [13]. Under a standard economic interpretation, the Lagrange multipliers of such techniques are referred to as *shadow prices* and can be shown to function as prices of network links [14]. The essential step in developing practical rate-control algorithms is to find a distributed algorithm for solving (1-2) in which each individual session need only compute a local optimization to set its own rates. There is a growing body of research devoted to finding such a distributed algorithm and using it as a basis for unicast rate-control in practical protocols [1,2,3,4,5,6,8,7, 15].

Observe that the topologies of sessions are not explicit in the formulation. For a unicast session, the links of $L(s)$ are arranged end-to-end, forming an acyclic path between a source and a receiver. However, $L(s)$ can be any subset of links—for example, a tree in the case of multicast. There is a requirement that the session employs a single rate x_s on all of its links. Thus, this formulation is readily generalized to single-rate multicast sessions.

[1] The terms 'session' and 'user' are synonymous in this paper.

The case of multi-rate multicast is somewhat more complicated since the singe rate requirement is replaced with a constraint that a session's rate on link l, be the maximum of its rate on any downstream link. Since a session can now have different rates on different links, it makes little sense to endow the session with a utility that is a function of scalar rate. Instead, recent treatments of the multi-rate problem [9,10] have altered the model by associating a utility function with each receiver. It is worth observing that this change to the model, while arising naturally from the multi-rate constraint, has been introduced without consideration of its effect on the global operating point. While a complete solution to the multi-rate problem is beyond the scope of this paper[2], our work provides a formal justification for the use of receiver utility functions even in the case of single-rate multicast where no such modelling pressure exists.

2.1 Application to Multicast

Single-rate multicast represents an important class of multicast applications. Many important applications, such as bulk data transfer [16] typically operate at a single rate. Even applications such as streaming video, for which multi-rate multicast is often considered well suited, single-rate multicast is used in current practice. It remains unclear whether multi-rate multicast for video is viable on the Internet, where it must be implemented using layered multicast schemes that have substantial overhead [17]. Furthermore, even if layered multicast is used, single-rate congestion control techniques may be appropriate to adapt the rates of individual layers.

It would appear that congestion control for single-rate multicast is a trivial extension of the unicast problem and can take advantage of existing approaches. It is important, however, to evaluate this claim carefully, given the importance of single rate multicast in practice. Certainly there are implementation issues in multicast that complicate the extension of unicast optimization-based rate control protocols based on packet marking schemes [5,6,4]. Equally serious, are the conceptual difficulties that arise in an uncritical application of the unicast solution of the underlying optimization problem. It is not immediately clear what the fairness properties of the resulting rate allocations are and, more fundamentally, what it means to define a utility function for a multicast session.

To develop our intuitions about the conceptual problems mentioned above, consider the approach of Low and Lapsley [5]. This approach finds a solution to problem (1-2) by solving its dual to obtain the following optimality condition for session rate x_s:

$$x_s(\lambda^s) = {U'_s}^{-1}(\lambda^s) \tag{3}$$

$$\lambda^s = \sum_{l \in L(s)} \lambda_l \tag{4}$$

[2] Solving the resulting multi-rate optimization problem is further complicated by the coupling of problem variables due to the multi-rate constraint and because the max function is nondifferentiable. See recent works by Kar, Sarkar, and Tassioulas [9] and Deb and Srikant [10] for treatments of the multi-rate problem.

where λ_l is the shadow price of link l and ${U'_s}^{-1}$ is the inverse marginal utility function for session s. It can be shown that ${U'_s}^{-1}(\lambda^s)$, and, hence, the rate allocated to session s is a decreasing function of the total session price λ^s. A large multicast session typically uses many more links than would a unicast session between the source and any single receiver and can therfore expect to see a higher session price than the unicast session. since ${U'_s}^{-1}(\lambda^s)$ is decreasing, multicast sessions will receive lower rates than unicast sessions along the same end-to-end paths, casting doubt on whether individual receivers have any incentive to join multicast groups. It may be reasonable to adopt a new definition of session utility with a bias in favor of multicast sessions to encourage bandwidth sharing. However, we must be careful not to overcompensate for the high session prices seen by multicast sessions, yielding allocations one would not consider fair to unicast sessions.

In the following sections we analyze the impact of such definitions on the fairness properties of the resulting congestion control mechansim. It will turn out that the class of receiver-oriented session utility functions, while not absolutely fair to unicast sessions, does not starve them in the presence of larger sessions. Moreover, we will see that utility functions in this class make sense in a way that other functions do not.

3 Multicast Utility Functions

In Section 2.1, we generalized the unicast optimization problem formulation to accommodate single rate multicast sessions. However, there is a subtle problem with this model that makes it difficult to apply to single-rate multicast. The problem concerns the definition of utility for an individual multicast session. An unweighted utility function is used to characterize the benefit of a higher rate to the session. For a unicast session, it makes little difference whether we consider this benefit to belong to the sender or receiver. For the purpose of unicast congestion control, we can treat the sender's and receiver's objectives as being one and the same. For a multicast session with multiple receivers it is unclear whether session utility belongs to the sender or should be split in some way among the receivers.

One approach towards defining multicast session utility ignores the multiplicity of receivers and defines it only with respect to the sender.[3] An alternative approach would be to define session utility as a function of the utilities of the receivers in the session. We informally refer to these two approaches as *sender-oriented* and *receiver-oriented*, respectively. While a receiver-oriented approach emerges naturally from the model in the case of multi-rate multicast, it is not immediately clear which approach is most appropriate for single-rate multicast. Later in this section we will formalize these definitions and argue in favor of a receiver-oriented approach. Before doing so, however, we will digress briefly to provide some background about the use of utility functions in economics and the theory of social choice.

[3] We are assuming that multicast sessions have a single source.

3.1 Utility Functions and Social Welfare

The use of concave increasing utility functions to represent session utility has a natural and intuitive interpretation. Utility is a monotonically increasing function of its input when individuals prefer having as much of the input as possible. The concavity of the utility function captures the idea of diminishing marginal utility[4]. Both concavity and monotonicity are appropriate assumptions in the case of bandwidth for elastic traffic [18], where the input to the utility function is the session rate.[5]

Utility can be difficult to quantify precisely; there is no clear unit of utility and no agreed upon scale. Comparing the utility of two individuals can be tricky, particularly when they do not share the same utility function. Because of the difficulty in performing interpersonal comparisons of utility, economists customarily think of utility as an *ordinal magnitude*, meaning that the absolute magnitude of utility is meaningless, but that the relative magnitudes of utilities at various rates for an individual session define preferences among rates and the relative differences in magnitude indicate the strength of the preferences [19]. A consequence of considering only ordinal magnitudes is that utility functions are unique only up to a linear transformation. That is, the utility maximizing behavior of an individual with utility function $u(x)$ is indistinguishable from one whose utility function is a linear transformation of $u(x)$. This restriction makes intuitive sense because a linear transformation simply represents a change in scale and a translation of the zero point of the utility function.

The notion of an aggregate utility function is a compelling extension of the concept of individual utility. Aggregate utility is defined by a *social welfare function* (SWF) that maps the vector of all session utilities to a scalar utility value representing the social desirability of the corresponding vector of rates. Since the SWF is not one-to-one, it induces a partial ordering over allocations of rates, known as the *social preference relation* (SPR). As with individual utility functions, we are primarily interested in this preference relation rather than the absolute magnitude of the SWF.

In optimization-based congestion control, we adopt the sum of individual utilities as the SWF. In general, there are many ways to define the SWF, each carrying with it some subjective judgment about how individual preferences should be combined to determine a social preference. It is possible to specify desirable properties for a SWF axiomatically. Perhaps the most important result of social choice theory is Arrow's Impossibility Theorem, which states that it is not always possible to satisfy all desidirata [20].

For example, a commonly cited property of SWFs is *independence of irrelevant alternatives*, which states that the socially preferred allocation should be

[4] The term 'marginal utility' is used in economics to refer to the first derivative of the utility function.

[5] In this section, utility will be assumed to be a function of session rate; we do so for the sake of concreteness and continuity with the rest of the paper. It should be understood, however, that the discussion presented here applies to any utility function.

invariant under a change in individual *utility functions* that leaves individuals' *preferences* unaffected. It is straightforward to show that the sum of individual utility functions violates this property. Indeed, it is precisely this violation that allows Kelly to associate optimal rates under different functional forms of utility with different formal definitions of fairness [1].

Although independence of irrelevant alternatives is neither required nor (in light of Arrow's Impossibility Theorem) worth pursuing for the congestion control application, a related but weaker property is still worthy of consideration.

– **Invariance Under Linear Transformation (ILT):** Let u be a vector of utility functions and v be a transformed vector such that $v_i(x) = \alpha\, u_i(x) + \beta$. Let $U(u(x))$ be a SWF, where $u(x) = (u_i(x_i))$ is the vector of session utilities for rate vector x. We say that a SWF is *invariant under a linear transformation* if, for any two rate vectors x and y,

$$U(u(x)) \geq U(u(y)) \Rightarrow U(v(x)) \geq U(v(y))$$

for any values of α, β. In words, the SWF induces the same preference relation for u and v.

The ILT property builds on the assertion that individual utility functions are unique up to a linear transformation, saying that aggregate preferences, too, should be invariant under such a transformation. We will see shortly that under some definitions of multicast session utility the ILT property is satisfied, while under others it is not.

3.2 Sender- and Receiver-Oriented Utility Functions

We now formally define sender- and receiver-oriented concepts of session utility. Consider a single-rate multicast session s with rate x and receiver set $\mathcal{R}$ with size R. In the sender-oriented approach, session utility function is a concave increasing function u_s of the session rate.

$$U_{snd} = u_s(x) \tag{5}$$

In the receiver-oriented approach, each receiver $i \in \mathcal{R}$ has a utility function $u_i(x)$, which is concave and increasing. The session utility function is the sum of receiver utilities.

$$U_{rcv} = \sum_{i \in \mathcal{R}} u_i(x) \tag{6}$$

We can convert these definitions into an alternate form by introducing two requirements. First, we require that all receivers in a session have identical utility functions.

$$u_i(x) = u_r(s) \quad \forall\, i \in \mathcal{R}$$

We typically think of utility functions as representing application characteristics and sometimes as being imposed by network mechanisms. For example, following the example of Kunniyur and Srikant [4], we use u(x) = -1/x to model TCP-style congestion control.[6] To the extent that receivers within a session share the same application requirements, it is also reasonable to assume they share a utility function. We feel that this is a natural assumption in the case of single-rate multicast. The second requirement is that both sender- and receiver-oriented utility functions should reduce to the same standard unicast utility function up to a linear transformation when $R = 1$. These two restrictions allow us to express both types of session utility functions as the product of a base utility function $u(\cdot)$ and a scaling function $f(\cdot)$. The base utility function, $u(\cdot)$ depends only on the session rate and is concave and increasing. It can be thought of as the utility function of a session with a single receiver. The scaling function $f(\cdot)$ depends on the number of receivers in the session. It must be monotonic in its argument, although it need not be strictly increasing.

For a sender-oriented definition of session utility, $f(R) = \kappa$, where κ is a constant.

$$U_{snd}(x, R) = \kappa\, u(x) \tag{7}$$

For a receiver-oriented definition, $f(R) = \kappa\, R$, where κ is a constant.

$$U_{rcv}(x, R) = \kappa\, R\, u(x) \tag{8}$$

It is possible to entertain other definitions of session utility. We choose these because they are commonplace and mathematically tractable. One obtains equation (7) by treating all sessions equivalently, regardless of the number of receivers. Equation (8) reflects the idea that multicast session utility is itself a social welfare function, representing the aggregate utility of the receiver set. Under our assumptions, this equation is equivalent to the sum of receiver utilities—a simple and commonly used social welfare function

3.3 The Session-Splitting Problem

In Section 3.2, we identified two alternative definitions of multicast session utility based on sender- or receiver orientation. Now we consider these two definitions in more detail and determine which makes sense in the context of congestion control. We begin by attempting to capture the effect of flexible group membership using an optimization-based approach. Golestani and Sabnani [23] observe that if receivers in a session can be dropped and reassigned to a different session in response to congestion, it is often desirable to split a multicast group into subgroups with different rates. One can think of this approach as an approximation of multi-rate multicast that does not violate the constraint of having a single rate per session and requires less overhead at the receiver [17].

The presence of additional sessions in the network after splitting may increase contention on existing bottlenecks or even create new bottlenecks. Thus not all

[6] A more accurate TCP utility function is $u(x) = (\sqrt{2}/T)\ tan^{-1}(T\,x/\sqrt{2})$, where T is the round trip time [21,22]. Kunniyur and Srikant's approximate function $u(x) = -1/x$ is valid for small end-to-end loss probabilites.

ways of splitting a session lead to an overall improvement in received rates. Ideally, one would like to find a way to split the session that offers a higher rate to some receivers without reducing the rates of any others. A less ideal, but perhaps still tolerable split might reduce some receivers' rates but improve the utilization of the network and allow many more receivers to receive at a higher rate. In this section, we consider the use of sender- and receiver-oriented social welfare functions to determine whether splitting a session will improve aggregate utility.

In general, the choice of sender- or receiver-oriented utility as well as the form of the base utility function will affect the social welfare function. However, for a fixed choice of these factors, we expect the SWF to be well-defined for all possible ways of splitting the session. Additionally, the optimal way of splitting a session should be invariant under a linear transformation of the base utility function. If this were not the case, an arbitrary rescaling of utility could determine whether splitting a session is preferred over not splitting. We will observe that this invariance holds in the case of a receiver-oriented SWF but not in the case of a sender-oriented one.

We begin by formalizing the session splitting problem in terms of utility maximization. In the session-splitting problem, we have a network (N, L) with link capacities $C = (c_l, l \in L)$. A set of receivers $R \subset N$ could be served by one or more multicast sessions with source $s \in N - R$. We assume that the number and rates of all other sessions in the network are fixed. Capacities in C thus represent the available capacity for multicast sessions serving receiver set R. Each session's rate is limited by its most constrained receiver, that is, by the receiver with the lowest link capacity along the path between it and the source. If this bottleneck link is not shared by all of the receivers, then it may be possible to split the session into two or more sessions yielding a higher rate to some receivers.

Splitting the session is equivalent to partitioning the receiver set into disjoint subsets $P = \{P_1, P_2, \ldots, P_N\}$. We will use $\mathcal{P}$ to denote the set of all possible partitions of R. Each partition in $\mathcal{P}$ represents one possible way to divide the receiver set into sessions. Each element of a partition represents a subset of R to be served by a different session. Rates may vary among sessions, but all receivers within a session must receive at a single rate. Computing the rates for each session is, itself, a non-trivial problem since some links will be shared by more than one session. There are many possible mechanisms for determining session rates. One example is the greedy algorithm suggested by Rubenstein, Kurose and Towsley [24] to achieve max-min fairness among the sessions.

For our purposes, it is sufficient to assume that we have some deterministic mechanism to perform this rate assignment, which we model as a rate allocation function $X : \mathcal{P} \times \mathbb{Z}^+ \to \mathbb{R}$. Given a partition P and an index i, the rate allocation function returns the rate of the session serving P_i.

The session-splitting problem requires us to find a partition that maximizes the aggregate utility of the network. Recall that the optimization-based approach defines aggregate utility as the (possibly weighted) sum of all session utilities.

Thus the optimal splitting is a partition that solves

$$\max_{P \in \mathcal{P}} U(P; f, u, X)$$

where

$$U(P; f, u, X) = \sum_{i=1}^{|P|} f(|P_i|)\, u(X(P, i))$$

is the aggregate utility function. We can choose the scaling function f from equations (7) and (8) to solve this problem for sender- and receiver-oriented definitions of session utility.

The aggregate utility function defines a partial ordering over $\mathcal{P}$. In economic terms, this ordering is the social preference relation over all possible partitions of the receiver set. As explained in Section 3.1, it is customary to regard utility functions as unique up to a linear transformation. A reasonable restriction, therefore, is only to allow social preference relations that remain invariant under a linear transformation of the base utility function, as captured by the following axiom, similar to the ILT property in Section 3.1:

Axiom 1 *Let $f(\cdot)$ be a fixed scaling function and $X(\cdot, \cdot)$ be a fixed rate allocation function. For any base utility fuction $u(\cdot)$, let $v(\cdot)$ be another base utility function such that*

$$v(x) = \alpha\, u(x) + \beta$$

where α and β are constants. Then for all $P, Q \in \mathcal{P}$,

$$\begin{aligned} U(P; f, u, X) \;&\geq\; U(Q; f, u, X) \\ \iff U(P; f, v, X) \;&\geq\; U(Q; f, v, X) \end{aligned}$$

Theorem 1. *Let $f_{snd}(R) = \kappa$ and $f_{rcv}(R) = \kappa\, R$ be sender- and receiver-oriented scaling functions. For any base utility function u and rate allocation function $X(\cdot, \cdot)$, the aggregate utility function $U(\cdot; f_{rcv}, u, X)$ satisfies Axiom 1, while $U(\cdot; f_{snd}, u, X)$ does not. Furthermore, Axiom 1 can only be satisfied using the scaling function $f(R) = f_{rcv}(R)$.*

The proof of Theorem 1 is quite straightforward and is omitted here due to space limitations. Interested readers can find it in [25]. One immediate consequence of this theorem is that if one accepts that Axiom 1 is indeed an appropriate requirement for any "reasonable" definition of aggregate utility, then our sender-oriented utility definition is not "reasonable". In fact, the only reasonable definition of session utility is a receiver-oriented one.

4 Consequences of Receiver-Oriented Utility

In Section 3.3 we argued that receiver oriented session utility functions are an appropriate model for multicast session utility in the session splitting problem.

In this section, we return to the original congestion control problem and determine whether using receiver-oriented utility functions leads to fair sharing of bandwidth between unicast and multicast sessions. We rewrite the network optimization problem (1-2) as

$$\max_{x=(x_s, s\in S)} \sum_s \kappa\, R_s\, u(x_s) \tag{9}$$

$$\text{subject to} \sum_{s\in S(l)} x_s \le c_l, \quad \forall l \in L \tag{10}$$

The Kuhn-Tucker conditions for optimality are

$$\kappa\, R_s\, du/dx = \sum_{l\in L(s)} \lambda_l \tag{11}$$

$$\lambda_l(x^l - c_l) = 0, \ (x^l - c_l) \le 0 \tag{12}$$

where the λ_l are Lagrange multipliers or link prices and $x^l = \sum_{s\in S(l)} x_s$ is the aggregate rate seen at link l. As before, we also write $\lambda^s = \sum_{l\in L(s)} \lambda_l$ as the total session price seen by session s.

From the first Kuhn-Tucker condition (11), we observe that the use of receiver-oriented utility functions creates a bias in favor of sessions with large numbers of receivers. To see this, note that

$$du/dx = \lambda^s\, (\kappa\, R_s)^{-1} \tag{13}$$

The optimal rate for session s, x_s^* is given by

$$x_s^* = u'^{-1}\left(\lambda^s\, (\kappa\, R_s)^{-1}\right) \tag{14}$$

Equation (13) states that, at optimality, the a session's marginal utility should be proportional to its price divided by the number of receivers. We refer to the ratio $\lambda^s/(\kappa\, R_s)$ as the *effective session price*. The optimal rate can therefore be obtained by taking the inverse of the marginal utility function as shown in (14). Since U_s is concave, u' is a strictly decreasing function of x and its inverse is also a decreasing function. For a fixed session price, a session with a larger number of receivers has a lower effective session price and thus receives a higher rate. We refer to this effect as "tyranny of the majority" (ToM).

ToM is a source of unfairness against unicast flows since multicast flows with the same total session price will receive a higher rate. However, the fact that multicast sessions tend to use more links than unicast sessions, particularly as the number of receivers becomes large, means that the session price λ^s for a multicast flow is likely to be higher than that of a unicast session. To understand the fairness properties of rate allocations under receiver oriented utility functions we must determine whether the price increase associated with the scaling of multicast trees is sufficient to limit the effect of ToM as more receivers are added.[7]

[7] If one holds that improving the rate of many receivers at the expense of a few is reasonable, giving a larger share of bandwidth to larger groups may not seem unfair.

5 Effect of Multiple Points of Congestion

In the previous section, we saw that ToM and the scaling of multicast trees have opposite effects. As we will see shortly, these effects are not necessarily equal in strength. The effect of ToM is likely to be the stronger of the two, allowing sessions with more receivers to receive a greater share of bandwidth. Whether we choose to accept this form of controlled unfairness or introduce a correction, we require a more precise understanding of the interaction of the two effects. In this section, we show that the functional form of the base utility function can be chosen to limit the strength of the ToM bias.

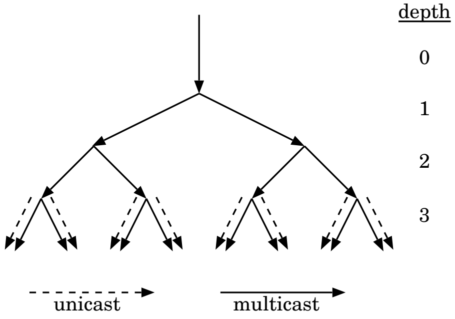


Fig. 1. A binary multicast tree of depth 3 with a sharing depth of 3.

Consider a multicast session in the form of a complete tree of degree k and depth D, such as the one shown in Fig. 1. Each link of the tree has capacity c. We will use a receiver-oriented definition of session utility, but allow an arbitrary base utility function $u(x)$. The tree has a receiver at each leaf, giving $R = k^D$ receivers in total. The multicast session shares the network with a set of k^D one-hop unicast sessions, which are evenly distributed accross the links at depth d. There are k^{D-d} unicast sessions on each of k^d links at level d. We refer to d, the depth in the multicast tree at which unicast sessions share links, as the *sharing depth*. Let $x = (x_s)$ be the vector of session rates, where x_0 is the multicast rate and $x_1, \ldots, x_R$ are the rates of the unicast sessions. Shadow prices are represented by a vector of multipliers $\lambda = (\lambda_1, \ldots, \lambda_L)$, where $L = k^d$.[8] For a particular choice of sharing depth d, we can now form the Lagrangian for the basic optimization problem (1).

We take the position that a bounded bias in favor of large groups is a defensible form of "controlled unfairness" but that there must be a mechanism to prevent starvation of unicast flows.

[8] The vector λ contains elements for only those links with nonzero price, namely, the k^d links at depth d.

$$\mathrm{L}_d(x, \lambda) = k^D\, u(x_0) + \sum_{i=1}^{R} u(x_i) - \sum_{j=1}^{L} \lambda_j\, g_j(x) \tag{15}$$

where $\{g_j\}$ is the set of capacity constraints for the shared links.

$$g_j(x) = x_0 + \sum_{l=L(j-1,k,d,D)}^{L(j,k,d,D)-1} x_l - c \leq 0 \tag{16}$$

$$L(j, k, d, D) = j\, k^{D-d}$$

We use the symmetry of the tree topology to reduce the problem to three variables: the multicast session rate x_m, the unicast session rate x_i, and the shadow price of a congested link λ. We rewrite the link capacity constraint

$$g_j(x) = g(x) = x_m + k^{D-d}\, x_i - c \tag{17}$$

Solving the first-derivative conditions of the reduced problem for the logarithmic base utility function $u(x) = \log(x)$ gives

$$x_m = c/2,\;\; x_i = c/(2\, k^{D-d}),\;\; \lambda = 2\, k^{D-d}/c \tag{18}$$

We observe the following facts about this result:

- At the system optimum, the multicast session receives rate $x_m = c/2$. This result is independent of the tree depth D, the sharing depth d, and the tree degree k.
- The invariance of the optimal multicast rate is a direct result of the choice of a logarithmic base utility function. As we will see, this property does not hold for other utility functions.
- The remaining capacity on the shared links is split evenly among the sharing unicast sessions. Since the number of sharing sessions is k^{D-d}, the optimal unicast rate depends on D, d and k.
- The total price charged to the multicast session is

  $$\lambda\, k^d = 2\, k^D/c,$$

  which is independent of the sharing depth. Under a receiver-oriented definition of session utility, this price is divided by the number of receivers to obtain the effective session price. Thus, effective session price is independent of d, D and k under a logarithmic utility function.

Since the invariance of the multicast session rate appears to derive from a special choice of utility function, it is interesting to explore the behavior as we modify the functional form. We can derive the following optimality condition:

$$u'(x_m) = \frac{1}{k^{D-d}}\, u'(\frac{c - x_m}{k^{D-d}}), \tag{19}$$

Equation (19) relates the marginal utility function $u'(x)$ to the function $u'^*(x) = a\,u'(a\,(c - x))$ obtained when we rotate u' about the line $x = c$ and scale both the argument and the result by the same factor a. Any point at which these two functions intersect satisfies the optimality condition. Note that $u'(x)$ is the derivative of a concave and strictly increasing utility function, and therefore must be strictly decreasing. Thus, $u'(x)$ and $u'^*(x)$ intersect in exactly one point, establishing the uniqueness of the solution. Observe also that the scaling factor $a = 1/k^{D-d} \leq 1$. Scaling the argument of $u'(c - x)$ compresses the function along the horizontal axis and moves the point of intersection to the left, while scaling its result compresses the function along the vertical axis and moves the point of intersection to the right.

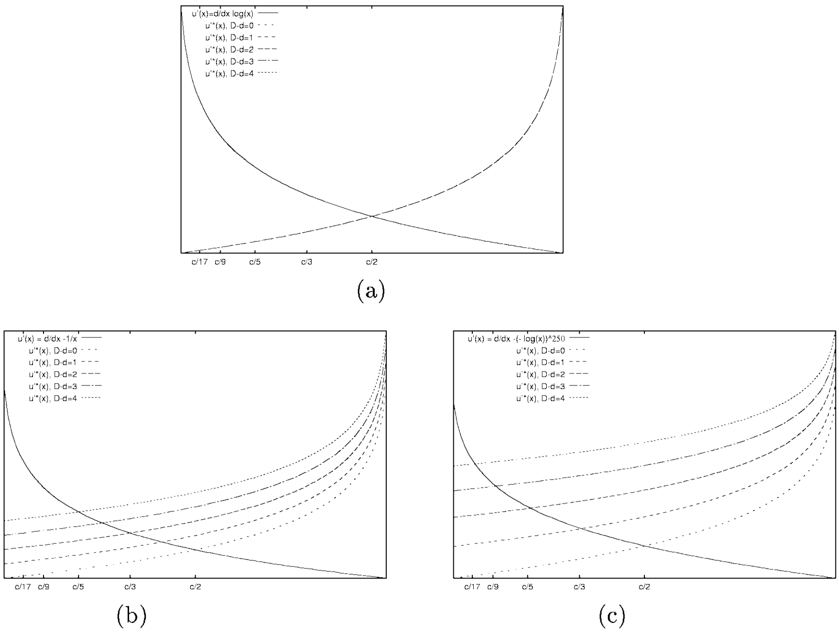
Figure 2 shows how the allocated multicast rate changes as we vary the sharing depth d (hence, the number of congested links) in a binary tree for three choices of base utility function: $u(x) = \log(x)$, $u(x) = -1/x$ and $u(x) = -(-\log(x))^{\alpha}$. The first function is the now familiar logarithmic utility function. The second is the *minimum potential delay* (MPD) utility function introduced by Massoulie and Roberts [26] and shown by Kunniyur and Srikant to model the utility of TCP traffic [4]. The third function is shown by Kelly to yield max-min fairness in the limit as $\alpha \to \infty$ [1].[9] In all three graphs, the single decreasing function is $u'(x)$, the first derivative of the base utility function, and the family of increasing functions are $u'^*(x)$ for decreasing a (increasing $D-d$). The points where $u'^*(x)$ intersects $u'(x)$ give the optimal rates for the multicast session as a fraction of available capacity.

As established above, the intersection point is invariant and equal to $c/2$ for logarithmic utility. The intersection point is also fixed at $c/2$ when $a = 1$ for all three functions, corresponding to a sharing depth equal to the maximum tree depth. In both the MPD and max-min fair utility functions, however, the intersection point moves to the left as a decreases. That is, as the sharing depth moves closer to the top of the tree, the *number* of bottleneck links decreases. However, as more unicast sessions share each bottleneck, the *price* on each congested link increases and the multicast session receives a smaller fraction of the available bandwidth.

Under the definition of max-min fairness for single-rate multicast [24], the multicast session must share bandwidth equally with all sessions on its most congested link. Thus, in the max-min fair allocation for our k-ary tree example, $x_m = c/(k^{D-d} + 1)$. In the case of Kelly's max-min fair utility function, we see that the optimal rates approach the max-min fair allocations, indicated by the tick-marks along the x-axis in Fig. 2. It appears that the points of intersection converge to these values as we transform the logarithmic utility function into the max-min fair utility function by increasing the exponent α. Demonstrating this convergence formally is somewhat difficult.

We can establish a similar result for a family of utility functions that includes both the logarithmic and MPD utility functions and also yields max-min fairness as a limiting case. Consider the family of utility functions $u(x)$ with first deriva-

[9] In our numerical analysis, we take α to a reasonably high power. ($\alpha = 250$)



(a)

(b)

(c)

Fig. 2. Figure showing the effect on the optimal allocation for a binary multicast tree as we vary the sharing depth. These graphs show three different marginal utility functions, $u'(x)$ along with their transformations $u'^*(x)$ for various choices of $D - d$ with the y-axis shown in log scale. The x-coordinate of the points of intersection give the optimal session rates as a fraction of available capacity. Max-min fair allocations for different values of $D - d$ are indicated along the x-axis. The figure shows that the logarithmic utility function (top) gives the multicast session half the available bandwidth regardless of the number of sharing unicast sessions, whereas the max-min fair utility function (bottom) splits bandwidth evenly among all sessions on the shared link regardless of the number of receivers. The MPD utility function (center) represents a compromise between these two extremes.

tives $u'(x) = 1/x^{\alpha+1}$. Such functions include $u(x) = \log(x)$, $u(x) = -x^{-\alpha}/\alpha$. This family of functions was originally identified by Mo and Walrand [7]. Members of this family are mathematically tractable since the functions $u'(x)$ are homogeneous, satisfying

$$u'(t\,x) = t^{-r}\,u'(x) \tag{20}$$

where $r = \alpha + 1$. We can simplify the optimality condition of (19).

$$x_m/\,(c - x_m) = a^{(1-r)/r} \tag{21}$$

As a further simplification, we can express the multicast rate as a fraction, p, of available capacity, $x_m = p\,c$. Solving for p, we get

$$p = \left(1 + a^{(1-r)/r}\right)^{-1}. \tag{22}$$

In the limit of large α, p converges to the max-min fair allocation.

$$\lim_{r\to\infty} \left(1 + a^{(1-r)/r}\right)^{-1} = a/(1+a) = 1/(k^{D-d}+1). \tag{23}$$

Following Kelly's example in [1], we can prove that $u(x) = -x^{-\alpha}/\alpha$ always gives max-min fairness in the limit $\alpha \to \infty$, by providing an absolute priority to smaller flows.[10] For two rates such that $x_{s^*} < x_s$,

$$u'(x_{s^*})/u'(x) = (x_s/x_{s^*})^{\alpha+1} \to \infty \quad \text{as } \alpha \to \infty$$

6 Fairness to Unicast Sessions

In Section 5, we observed that a multicast session was able to obtain a higher rate than unicast sessions sharing the same bottleneck links. We showed that this unfairness is bounded in the presence of multiple points of congestion. However, this result exploited features of an idealized multicast session topology. Adopting a somewhat more realistic model in this section, we investigate whether the same type of bounded unfairness is possible in a more general setting with receiver-oriented utility functions. We also consider whether there is any multicast utility function that allows a strictly equal split of shared bottleneck bandwidth between a multicast an unicast session.

Adopting the fairness objective proposed by Handley, Floyd and Whetten [28]—that the algorithm be provably fair relative to TCP in the steady state, we define a generalized notion of TCP fairness. We say that a multicast session utility function $U(x;r) = f(R)\,u(x)$ is *strictly unicast-fair* if the optimal rate for the multicast session is the same as would be obtained by a unicast session with utility function $u(x)$ along the most congested source-to-receiver path in the multicast tree. This definition is equivalent to TCP-fairness in the case where $u(x) = -1/x$, the MPD utility function.

We first show that neither sender nor receiver oriented multicast utility functions lead to strict unicast-fair allocations and derive a result suggesting that strict fairness is difficult to achieve under any definition of session utility. Consider the modified star network topology shown in Figure 3. A single multicast session with source node s and receivers $\{1, \ldots, R\}$ shares the network with R unicast sessions, one from s to each receiver. Link l_0 from the source to the central node is shared by all sessions and has effectively infinite capacity. Each link l_i from the center to receiver i is shared by the multicast session and one unicast session. Link l_1 is the bottleneck link, with capacity $\beta\,c$, where $\beta < 1$ and c is the capacity of all other links $l_i,\ i > 1$. Receiver 1 is the most congested receiver in the multicast session.

[10] A similar result is reported in recent work by Bonald and Massoullie [27].

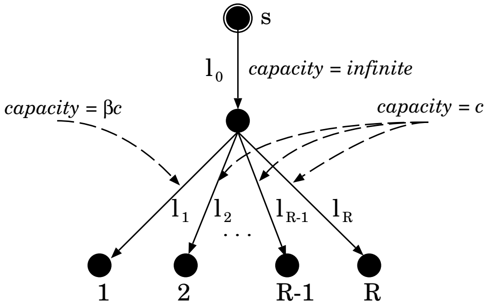


Fig. 3. A multicast tree with a modified star topology. Receiver 1 is most congested.

We give the unicast sessions the MPD utility function $u(x) = -1/x$. The multicast function has utility function $u(x; R) = f(R)\, u(x)$. Let x_m be the rate of the multicast session and x_i be the rate of the unicast session to receiver i. A strictly tcp-fair allocation would split the bandwidth on l_1 equally between x_m and x_1, $x_m = x_1 = \beta\, c/2$. We can substitute this rate into the optimality conditions of the optimization problem (9-10) to determine the appropriate scaling function $f(R)$ that will lead to the tcp-fair allocation, obtaining

$$f(R) = 1 + \frac{(R-1)\,\beta^2}{2\sqrt{2}\,(2-\beta)^2} \tag{24}$$

This result shows that tcp-fairness can be achieved in the optimization-based framework by maximizing a weighted sum of utilities with weights given by a scaling function $f(R)$. However, the presence of β, a topological parameter, in the scaling function suggests that the correct scaling function depends on topological properties of the network.

We now consider a generalized version of the previous example with no explicitly defined network topology. Consider a network containing a set of links $\mathcal{L}$. The network is shared by two sessions v and w, which have rates x_v and x_w, respectively. Each session uses a subset of links in the network and session w only uses a proper subset of links that are also used by v. Formally, $L(w) \subset L(v) \subseteq \mathcal{L}$. The sessions have R_v and R_w receivers with $R_v > R_w$. We assume that the path to the most constrained receiver in both v and w is the same and is therefore entirely contained in $L(w)$. The Lagrangian for the optimization problem is.

$$\begin{aligned} \mathrm{L}(x;\lambda) = {} & f(R_v)\, u'(x_v) + f(R_w)\, u'(x_w) + \\ & \sum_{l \in L(w)} \lambda_l (x_v + x_w - c_l) \;\; + \sum_{l \in L(v)-L(w)} \lambda_l (x_v - c_l) \end{aligned} \tag{25}$$

From the Kuhn-Tucker conditions, we derive an optimality condition on the ratio of marginal utilities.

$$u'(x_v)/u'(x_w) = (f(R_w)\,\lambda^v)/(f(R_v)\,\lambda^w) \tag{26}$$

where

$$\lambda^v = \sum_{l \in L(v)} \lambda_l, \quad \lambda^w = \sum_{l \in L(w)} \lambda_l \tag{27}$$

Consider the family of base utility functions satisfying $u'(x) = -1/x^\alpha, \ \ \alpha \geq 1$. Note that this family includes both the MPD and logarithmic utility functions. The ratio of session rates is

$$x_w/x_v = ((f(R_w)\,\lambda^v)/(f(R_v)\,\lambda^w))^{1/\alpha} \tag{28}$$

In a strictly tcp-fair allocation, the ratio $x_w/x_v = 1$. From equation (28), it is clear that the actual value of this ratio depends on both the choice of scaling function and the ratio λ^v/λ^w. It is also apparent that the ratio x_w/x_v approaches 1 in the limit as $\alpha \to \infty$. Thus, the exponent α offers one way to control unfairness for any choice of scaling function; increasing it moves the resulting rate allocation closer to max-min fairness. However, only the max-min utility function can guarantee strict unicast fairness for an arbirtary choice of scaling function.

If a utility function other than max-min is used, providing strict unicast fairness requires careful selection of the scaling function. For example, strict unicast fairness could be achieved by exploiting a scaling law relating the total price of a multicast session to its number of receivers. Chuang and Sirbu propose such a law for static multicast costs [29] with the form

$$\lambda^s \propto R_s^k \tag{29}$$

The authors empirically evaluate the scaling exponent k, finding its value to be constant over a wide range of network topologies. This law assumes, however, that link costs in the network are static. To be applicable for the purposes of congestion control, such a scaling law would have to be established for dynamically changing prices that reflect link congestion. If such a scaling law can be found, then strict unicast fairness would result from a multicast session utility function

$$U_s(x) \propto R_s^{-k}\, u(x).$$

We leave the search for such a scaling law as direction for future research, but note here that, as presented in Section 3.3 the sum of session utilities under such a multicast utility function would not be invariant under a linear transformation of $u(x)$.

7 Conclusion

This paper presented an optimization based scheme for multicast congestion control based on utility maximization. Appealing to the economic theory underlying this approach, we proposed the use of a receiver oriented definition of session utility. By considering the incentive to split multicast sessions into smaller sessions, we showed that only receiver oriented utility functions ensure that the optimal solution of the session-splitting problem remains invariant under a linear transformation of the utility scale.

We identified two sources of unfairness that arise when maximizing the sum of receiver oriented utility functions, one favoring unicast sessions and one favoring multicast. Unicast sessions are favored because they tend to use fewer links than multicast sessions and thus are charged a lower price for bandwidth. Multicast sessions are favored by the tyranny of the majority effect because the the sum of link prices in the session session is divided by the number of receivers and this reduced price is used to compute the session rate. When these two effects are combined, a net unfairness results that favors sessions with many receivers over sessions with few, with unicast sessions faring worst of all. This unfairness can be controlled, however, by choosing the form of the base utility function. While it is difficult to achieve strict fairness between unicast and multicast traffic, we argue that controlled unfairness is a reasonable goal, particularly as it provides an incentive to use multicast by rewarding larger groups.

Much work still remains to be done in this area. In the work presented here, we have focused on the economic interpretation of the optimization-based approach to reason about the fairness properties resource allocation at system equilibrium. A complimentary line of enquiry concerns the convergence and stability properties of the equilibria in the multicast case. A promising direction of future work is to extend the growing body of relevant research for unicast [30, 31,32,33] to the multicast case.

References

1. Kelly, F.: Charging and rate control for elastic traffic. European Transactions on Telecommunications **volume 8** (1997) 33–37
2. Kelly, F.P., Maulloo, A., Tan, D.: Rate control in communication networks: shadow prices, proportional fairness and stability. Journal of the Operational Research Society **49** (1998) 237–252
3. Gibbens, R., Kelly, F.: Resource pricing and the evolution of congestion control. Automatica **35** (1999) 1969–1985
4. Kunniyur, S., Srikant, R.: End-to-end congestion control: utility functions, random losses and ecn marks. In: Proc. INFOCOM. (2000)
5. Low, S.H., Lapsley, D.E.: Optimization flow control, i: Basic algorithm and convergence. IEEE/ACM Transactions on Networking (1999)
6. Athuraliya, S., Laspsley, D., Low, S.: An enhanced random early marking algorithm for internet flow control. In: Proc. INFOCOM. (2000)
7. Mo, J., Walrand, J.: Fair end-to-end window-based congestion control. IEEE/ACM Transactions on Networking (1999)
8. Golestani, S., Bhattacharyya, S.: A class of end-to-end congestion control algorithms for the internet. In: Proc. ICNP'98. (1998)
9. Kar, K., Sarcar, S., Tassiulas, L.: Optimization based rate control for multirate multicast sessions. In: Proceedings of Infocom 2001. (2001)
10. Deb, S., Srikant, R.: Congestion control for fair resource allocation in networks with multicast flows. In: Proc. of the IEEE Conference on Decision and Control. (2001)
11. Kasera, S., Bhattacharyya, S., Keaton, M., Kiwior, D., Kurose, J., Towsley, D., Zabele, S.: Scalable fair reliable multicast using active services. IEEE Networks Magazine (2000)

12. Rizzo, L.: pgmcc: A TCP-friendly single-rate multicast. In: Proceedings of SIGCOMM. (2000) 17–28
13. Madden, P.: Concavity and Optimization in Microeconomics. Basil Blackwell (1986)
14. Hillier, F.S., Lieverman, G.J.: Introduction to Mathematical Programming. 2 edn. McGraw-Hill (1995)
15. P.Key, McAuley, D., Barham, P., Laevens, K.: Congestion pricing for congestion avoidance. Technical Report MSR-TR-99-15, Microsoft Research (1999)
16. Byers, J.W., Luby, M., Mitzenmacher, M., Rege, A.: A digital fountain approach to reliable distribution of bulk data. In: Proc. Sigcomm '98. (1998)
17. Li, X., Ammar, M.H., Paul, S.: Video multicast over the internet. IEEE Networks Magazine (1999)
18. Shenker, S.: Fundamental design issues for the future internet. IEEE J. Selected Areas Comm. **13** (1995) 1176–1188
19. Hirshleifer, J., Hirshleifer, D.: Price Theory and Applications. 6 edn. Prentice Hall (1997)
20. Arrow, K.J.: Social Choice and Individual Values. 2 edn. Yale Univ. Press (1963)
21. Kelly, F.P.: Mathematical modelling of the internet. In: Proc. 4th International Congress on Industrial and Applied Mathematics. (1999)
22. Low, S.H.: A duality model of tcp flow controls. In: Proc. ITC Specialist Seminar on IP Traffic Measurement, Modeling, and Management. (2000)
23. Golestani, S.J., Sabnani, K.K.: Fundamental observations on multicast congestion control in the internet. In: Proc. INFOCOM. (1999)
24. Rubenstein, D., Kurose, J., Towsley, D.: The impact of multicast layering on network fairness. In: Proc. SIGCOMM 99. (1999)
25. Shapiro, J.K., Towsley, D., Kurose, J.: Optimization-based congestion control for multicast communications. Technical Report UM-CS-2000-033, University of Massachusetts at Amherst (2000)
26. Massoulie, L., Roberts, J.: Bandwidth sharing: Objectives and algorithms. In: Proc. INFOCOM. (1999)
27. Bonald, T., Massoullie, L.: Impact of fairness on internet performance. In: Proc. ACM SIGMETRICS. (2001)
28. Handley, M., Floyd, S., Whetten, B.: Strawman specification for tcp friendly (reliable) multicast congestion control. Technical report, Reliable Multicast Research Group (1998)
29. Chuang, J., Sirbu, M.: Pricing multicast communications: A cost-based approach. In: Proc. INET'98. (1998)
30. Low, S.H., Paganini, F., Doyle, J.C.: Internet congestion control. IEEE Control Systems Magazine (2002) February
31. Johari, R., Tan, D.: End-to-end congestion control for the internet: Delays and stability. To appear in IEEE/ACM Transactions on Networking (2001)
32. Massoulie, L.: Stability of distributed congestion control with heterogeneous feedback delays. Technical report, Microsoft Research (2000)
33. Hollot, C., Misra, V., Towlsey, D., Gong, W.: On designing improved controllers for aqm routers supporting tcp flows. In: Proc. IEEE Infocom 2001. (2001)

Severe Congestion Handling with Resource Management in Diffserv on Demand

András Császár[1,2], Attila Takács[1,2], Róbert Szabó[1,2],
Vlora Rexhepi[3], and Georgios Karagiannis[3]

[1] *N*et*L*ab, Ericsson Research HUNGARY
{robert.szabo, andras.csaszar, attila.takacs}@eth.ericsson.se
[2] High Speed Networks Laboratory,
Department of Telecommunications and Telematics,
Budapest University of Technology and Economics,
{robert.szabo, andras.csaszar, takacs}@ttt.bme.hu,
[3] ELN, Ericsson EuroLab Netherlands
{vlora.rexhepi, georgios.karagiannis}@eln.ericsson.se

Abstract. Quality of Service (QoS) for the Internet has been discussed for a long time without any major breakthrough. There are several reasons, the main one being the lack of a scalable, simple, fast and low cost QoS solution. A new QoS-framework, called resource management in differentiated services (RMD), aims to correct this situation. This framework has been published in recent papers and is extending the IETF differentiated services (diffserv) architecture with new admission control and resource reservation concepts in a scalable way. This paper focuses on proposing and investigating two resource reservation solutions on the problem of severe congestion situation within a diffserv-aware network utilizing an admission control scheme called Resource Mananagement in Diffserv (RMD). The different severe congestion solutions are compared using extensive simulation experiments.

1 Introduction

Internet QoS has been the most challenging topic of the networking research for several years now. The two existing Internet Protocol (IP) quality of service (QoS) architectures, Integrated Services (intserv) and Differentiated Services (diffserv) [1] are the results of the research work in this area.

Currently, the increasing popularity of the Internet as well as the growth of mobile communications have driven the development of IP-based solutions for wireless networking. The introduction of IP-based transport in radio access networks (RANs) is one of these networking solutions. When compared to traditional IP networks, an IP-based RAN has specific characteristics (see e.g. [2]) that impose stricter requirements on resource management schemes. Independently of the transport network, the cellular user expects to get the same service as in STM-based transport networks. In addition to this requirement, the situation is further complicated by the fact that the RAN is large in terms of its

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 443–454, 2002.

geographic size and the number of inter-connected nodes (hundreds or even thousands of nodes) with high cost of leased transmission lines, and the proportion of real-time traffic may get up to 100%. Resource management and CAC schemes working in IP-based RANs will have to enable dynamic admission control, fast resource reservation and at the same time they need to be simple, have low cost and easy to implement along with good scalability properties.

This paper focuses on proposing and investigating with simulation experiments two resource reservation solutions on the problem of severe congestion situation within a diffserv-aware network utilizing an admission control scheme called Resource Mananagement in Diffserv (RMD) [3] described in the following section. Severe congestion can be considered as an undesirable state, which may occur as a result of a route change or a link failure. Typically, routing algorithms are able to adapt to reflect changes in the topology and traffic volume. In such situations the re-routed traffic will traverse a new path. Nodes located on this new path may become overloaded, since they suddenly might need to support more traffic than their capacity. Moreover, when a link fails, traffic passing through it may be dropped, degrading its performance.

The rest of the paper is organized as follows: Section II lists related work in the resource management field. The severe congestion requirements and solutions are described in Section III. Section IV presents the simulation experiment results and their analysis. Finally, Section V concludes.

2 Related Works

Resource provisioning and traffic control algorithms use a signaling protocol to communicate the resource needs from end systems to routers, which either rely on information collected by measurements [4,5] or maintain some sort of reservation state. Generally, one group of approaches requires from every network entity to maintain per-flow state related information [6,7]. Another broad class of algorithms does not require per flow state related information, but is rather maintaining aggregated states in network core nodes, e.g., [8]. These mechanisms generally assume soft reservation status in the network, and either aim to periodically update it or try to harmonize the actions of routers along the path or take an economic approach to handle congestion [9].

2.1 Resource Management in DiffServ – RMD Framework

Currently, none of the available existing approaches satisfy the requirements for an appropriate resource management scheme within an IP-based RAN. In several recent papers [10,11] and IETF drafts [3,12] a new QoS framework, called Resource Management in DiffServ (RMD), is specified that aims to correct this situation. RMD extends the diffserv architecture with dynamic admission control and resource provisioning, and has good scaling properties and as such has low cost of implementation. Moreover, this framework has a wide scope of applicability in different types of diffserv networks.

In compliance with diffserv concepts, the RMD framework distinguishes between the problem of a complex reservation within a domain and handling a simple reservation within a node. Accordingly, there are two types of protocols defined within the RMD framework, the Per Domain Reservation (PDR) and Per Hop Reservation (PHR) protocol groups. *Per Domain Reservation* (PDR) is implemented only at the edges of the RMD domain and it handles the resource management in the entire diffserv domain. *Per Hop Reservation* (PHR) is used to perform resource reservation per diffserv class or Per Hop Behaviour (PHB) in each node of the diffserv domain. PHR aware nodes are not able to differentiate between individual traffic flows, as for e.g., RSVP, because no per-flow information is stored and the packet scheduling is done per aggregate. This way, PHR is optimized to reduce the functionality requirements of interior nodes.

In the following, we describe the simplified PHR operation. Before a new user data flow is admitted into the domain on one of the ingress edge nodes, it first has to signal its resource requirement (*QoS Request*). The ingress node classifies it into an appropriate PHB. These resource requests are transformed to discrete bandwidth values. Then the ingress edge node sends a *PHR Resource Request* packet to the egress edge, which is marked by any of the intermediate routers if they have not enough free resources. The egress edge node reports the reservation status back to the ingress, as a result of which the ingress can admit or reject the QoS request. If the flow is admitted, then periodic reservation refreshes are sent between the ingress and egress edge nodes.

The RMD framework [3] defines two different PHR groups: the reservation-based and the measurement based groups, which differ in the method a core node determines whether to mark a resource request packet, along with some signaling needs for this purpose. Here, we solely focus on the reservation-based PHR methods, where nodes maintain a per PHB reservation state. This is accomplished by using a combination of the reservation soft state and the explicit release principles. This means that the reserved resources can be released either when they are not refreshed regularly (1 refresh packet in every *PHR refresh period*), or when they are explicitly released by a predefined release message. In order to decrease the signaling traffic load on the network, the number of PHR refresh messages has to be minimized. Therefore, the PHR refresh period has to be chosen as large as possible, e.g., 30 seconds. The admittance decision is based on a threshold of maximum available resource units set for each PHB. Currently, there is one reservation-based PHR protocol defined, the Resource Management in Diffserv On DemAnd (RODA) protocol specified in [12].

3 Severe Congestion

3.1 Problem Definition and Requirements

Severe congestion can be considered as an undesirable state that may occur as a result of a route change or a link failure. The severe congestion situation will severely degrade the performance of the real time traffic and therefore, it has to be detected and solved very fast. Typically, in a RAN where majority of traffic is real-time traffic, the severe congestion situation has to be detected by the ingress edges within one second. Subsequently, the ingress edge has to undertake predefined policing actions to lower the incoming traffic volume in order to solve the severe congestion situation. The severe congestion solution can be decomposed in four subsequent phases:

- *Detection of severe congestion by interior nodes:* An RMD interior node has to detect the severe congestion situation using one of the following methods:
  - Volume measurements: by using measurements on the data traffic volume. If the volume of the data traffic increases suddenly, it is deduced that a possible route change and at the same time, a severe congestion situation occurred.
  - Counting: using a counter that counts the number of dropped data packets. The severe congestion state is activated when this number is higher than a pre-defined threshold. This method is similar to the previous one but is much simpler. However, it can only be applied when the traffic characteristics are known.
  - Increased number of refreshes: if the number of resource units per PHB, refreshed by PHR refresh messages is much higher than the number of resources refreshed previously, then the node deduces that a severe congestion occurred. This procedure is very efficient, but it can only be used when the PHR refresh period is small.

  The first three detection methods can be applied on both RMD schemes, i.e., reservation-based and measurement-based. The last method can only be applied on the reservation-based RMD scheme.

- *Propagation of severe congestion state to egresses*: In this phase, an interior node notifies the severe congestion situation to an egress node. Due to the fact that the interior node does not store and maintain any flow related information, it is not possible to identify the ID of the passing flow and the IP address of the ingress node. Therefore, the interior node is not able to directly notify the ingress node that a severe congestion situation occurred. One of the following methods is applied:
  - Greedy marking: all packets which are passing through a severe congested interior node and are associated to a certain PHB will be somehow remarked to indicate severe congestion;
  - Proportional marking: this method is similar to the previous method, with the difference that the number of the remarked packets is proportional to the detected overload;
  - PHR message marking: only PHR signaling messages that are passing through a severe congested interior node will be marked. The marking is done by setting a special flag in the protocol message, i.e., "S" (see [12]). This procedure is efficient, but it can only be used when the PHR refresh period is small.

  The last method can only be applied on the reservation-based scheme, while the other two can be applied on both RMD schemes.

- *Egress to ingress state propagation:* In this phase, the egress node has to process the severe congestion information received from the interior nodes. Moreover, it notifies the ingress node that a severe congestion occurred. The type of the received severe congestion information depends on the propagation method used by the interior nodes (see above). When either the “greedy marking” or the “PHR message marking” method is used, the severe congestion information simply notifies the egress node that a severe congestion situation occurred. When the “proportional marking” method is used, the egress node is informed that a percentage of the incoming traffic is overloading a certain communication path. The egress node forwards this information to the proper ingress node for all congestion marked flows.

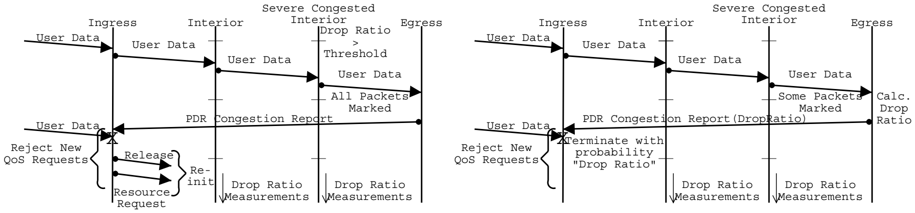
- *Ingress actions on severe congestion:* the ingress node processes the severe congestion information received from the egress node and undertakes certain actions to solve the severe congestion situation. These actions depend on the method used by the interior nodes. One of the following set of actions can be undertaken:
  - Re-allocation: an ingress node is blocking new traffic flows and re-initiates all on-going flows that are affected by severe congestion. During the re-allocation procedure the ingress nodes will temporarily release and subsequently re-initiate all on-going flows that were affected by severe congestion.
  - Stochastic blocking: an ingress node is blocking new traffic flows and is terminating some of the on-going flows based on a probabilistic competition. The termination probability of a connection is proportional to its severe congestion marked traffic volume.

  The first method is applied when either the “greedy marking” or “PHR message marking” procedure is used. The later can only be applied when the “proportional marking” detection procedure is used.

3.2 Approaches to Handle Severe Congestion

In all of the consecutive approaches we commonly assume the followings: *i*) interior nodes use the *counting* detection method. In particular each interior node performs packet drop ratio measurements for every S interval per DSCP; *ii*) ingress nodes maintain per flow information that includes the flow ID, the requested amount of resources, i.e., bandwidth units; *iii*) egress nodes maintain per flow information that includes the flow ID and the IP address of the ingress node; *iv*) each node is capable of remarking each standardised DSCP, into locally defined DSCPs in order to signal severe congestion; *v*) egress nodes are checking the DSCP field of each passing data packet in order to identify its severe congestion status.

Solution A – Re-allocation of resources. The main characteristics (Fig. 1) of this scenario are that each interior node uses the *greedy marking* procedure and



Solution A

Solution B

Fig. 1. Solution A and B Operation

each ingress node uses the *re-allocation* procedure to solve the severe congestion situation. Whenever the measured drop ratio in an S interval is higher than a pre-configured threshold value, the interior node deduces that a severe congestion occured. It remarks the DSCP field of all passing packets into a locally defined DSCP to indicate severe congestion. The egress nodes monitor the DSCP fields of each passing data packet. If re-marked DSCP fields are detected, then the egress node will deduce that a severe congestion occured. For each affected flow, the egress node will report the severe congestion situation to the corresponding ingress node by using a signaling PDR congestion report message.

When the ingress node receives a PDR congestion report message from the egress, it will block new incoming flows for a certain amount of time. Moreover, the on-going flows that are affected by severe congestion have to re-initiate their reservations: they must temporarily terminate their user data flow, deallocate their reservation with an explicit *release request* message, and try to reallocate their original resource usage along the path. Note that the *release request* and *reservation request* messages are not necessarily sent immediately after congestion notification as we will show later on, though user data must immediately be terminated.

Interior nodes receiving a *release request* message, decrease their aggregate reservation states with the number of units indicated in the corresponding message but not below zero. Unfortunately, in the case of a link failure and re-route event, flows previously accommodated on different paths will try to release resources on links where they have not yet allocated. In order to cope with this problem, interior nodes are disallowed to release more resources than previously allocated. Upon receiving a *reservation request* message, the interior node must see whether the sum of already existing reservations plus the new request is within a threshold. If so, the interior node updates its reservation state or else it marks the packet to indicate reservation failure.

The new reservation might be admitted by all interior nodes and signaled back by the egress to restart the user data. However, if the re-initiated allocation fails or time-outs (one may allow several retries before permanently terminating any connections) then the connection must be terminated by the end hosts.

Therefore, the *ingress node* must initiate the final termination of the flow at the end host.

Solution B – Stochastic (distributed) blocking. The main characteristics (Fig. 1) of this scenario are that each interior node uses the *proportional marking* procedure and each ingress node uses the *stochastic blocking* procedure to solve the severe congestion situation.

When the *ingress node* receives the PDR congestion report message, it will block new incoming flows for a certain amount of time. Moreover, it will also terminate some of the ongoing connections: it will realize a probabilistic drop on each individual flow that has received congestion report message according to the following formulas:

- Algorithm B1: $P_{\text{drop}}^{\text{B1}} = \frac{\text{\# of marked bytes}}{\text{\# of marked bytes}+\text{\# of unmarked bytes}}$. The underlying idea here is to purely base the blocking estimation on measured data. In this algorithm the blocking probability per connection (per flow) is calculated as the ratio between the dropped bytes and the maximum number of bytes that can be supported by the interior node (dropped / received).
- Algorithm B2: $P_{\text{drop}}^{\text{B2}} = \frac{\text{\# of marked bytes}}{rS_e 8}$, where r [bps] is the allocated rate for the connection, S_e [sec] is the time base used at the egress edge and 8 is for bit/byte conversion. This version aims to eliminate the packet drops of connections by using the administrated reservations (dropped / sent-administrative).
- Algorithm B3: $P_{\text{drop}}^{\text{B3}} = \frac{\text{\# of marked bytes}}{2*\text{\# of marked bytes}+\text{\# of unmarked bytes}}$. Here, the blocking probability per connection (per flow) is calculated as the ratio between the dropped bytes and the total volume of user data (associated to the same connection), entering the RMD domain (dropped / sent-measured).

4 Numerical Results

In this paper we compare the severe congestion solutions described in Section 3 by using performance evaluation with the help of simulations. For these simulation experiments we used the network simulator (ns) [13] environment. This section describes the used traffic models, network topology, the performed experiments and their results.

4.1 Simulation Model

Traffic models: Based on the operational description of the RMD protocol, resource demands are handled in bandwidth units, which we will also use when describing our traffic models. First of all, one unit was set to represent 2000 bytes/second rate allocation. The reason behind was that one unit should represent the rate required by an encoded voice communication, e.g., GSM coding. Altogether we examined three different scenarios where i) calls requested only 1 units homogeneously, where ii) calls requested bandwidths of $\{1, 2, \ldots, 20\}$

units and iii) where call demands were selected from $\{1, 2, \ldots, 100\}$ units. Calls arrived according to a Poisson process with parameter λ_i for calls requesting i units of bandwidth. The average call holding time was set to $\frac{1}{\mu} = 90$ seconds. The call and bandwidth unit requests were generated in a way that the demanded load for each reservation unit class was eqalized on the average, or more formally $\frac{\lambda_i}{\mu} BW_i = \frac{\lambda_j}{\mu} BW_j$, where $\mathrm{BW}_i = i$[unit] is the bandwidth demand and $\frac{1}{\lambda_1} = 0.9$ sec as per default. Hence higher bandwidth demands arrived less frequently than smaller ones. This is not unrealistic since higher bandwidth requests are probably more expensive. Packet sizes (L) of the connections were determined according to their reservations in the following way: $L_i = 40$ [bytes]BW_i. For the sake of simplicity packet inter arrival times were kept constant (constant bit rate - CBR), hence every flow sent one single packet in each 20 msec interval.

Network topology: For the evaluation of the methods we used a simple delta topology with the $G(V, E)$ graph, where E denotes edges $\{e0, e1, e2\}$ and V denotes the vertices $\{(e0, e1), (e0, e2), (e1, e2)\}$. With the former re-routing and its effects are easily traceable. For the sake of simplicity only $e0$ generated traffic for the other two edges. In order to have effective multiplexing of flows, the capacity (C) of the links was set to be able to accommodate at least 100 flows of the highest bandwidth demands, i.e., to 100, 2‘000 and 100‘000 units.

As discussed earlier, the severe congestion detection is based on packet drop ratio measurements. Hence, it was important to find the proper dimensioning for the network buffers. As our traffic model was based on CBR traffic with 20 msec packet inter arrival times, we determined the queue lengths so that no packet loss can occur during normal operation. We dimensioned for a target load level of 80% link capacity, hence the buffer sizes (B) were determined using the following formula: $B = C * 0.02 * 0.8$ [bytes].

Network events: In our simple delta network after the system achieves stationarity, the link between nodes $e0$ and $e2$ goes down at 350 sec of simulation time. Afterwards, the dynamic routing protocol (OSPF) updates its routing table at 352.0 sec and all flows previously taking the $e0 - e2$ path will be re-routed to the $e0 - e1 - e2$ alternate path. Hence, node $e0$ will suffer a serious overload resulting from the re-route event.

4.2 Numerical Evaluation

Network utilization: It can be seen in Fig.2/a that with solution A, more reservation messages were accepted by the severe congested node during the re-initiation procedure than the target admission threshold (80%). This phenomenon is due to certain properties of the protocol related to the explicit release of soft state resources, whose basic idea is discussed in [11]. Nevertheless, the above problem can briefly be reasoned by the following operation:

First of all, in order to achieve better utilization the soft state refresh period (T), which is in all cases set to 30 sec, is sub-divided into cells (10), where a sliding (or time) window algorithm is used to smooth out the T long discrete time steps. If a re-route event happens in the system after the link failure, all of the traffic originally traversing along a different path will flow through the single operating link. This will evidently make severe congestion situation. Here however, not only data packets but protocol signaling packets (see [12]) are involved, which will affect the administrated reservations in the following way. With solution A, release messages of the re-initialization procedure (see section 3.2) will decrease the number of registered reservations, however not only the originally accommodated flows but also the re-routed ones will try to release their allocations. Hence the volume of release must be limited, which is done by permitting releases until the administered reservation state is above zero. Unfortunately, connections that have not yet been notified of severe congestion (longer round trip times (RTT)) will keep on sending their periodic reservation refreshes that increase the administered reservation. This affects the same reservation state that the release messages compete for, hence allowing more release and reallocation than desired. This overshoot will only leave the system after a refresh period (see it at around 382 sec in Fig. 2/a).

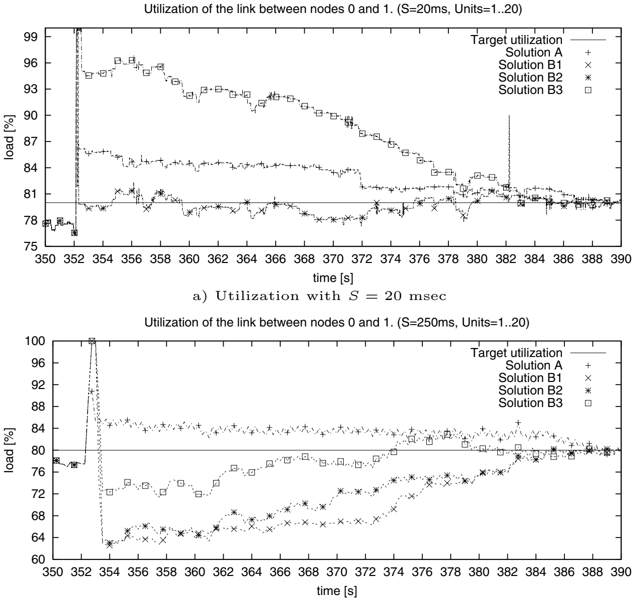
On the other hand, the descendant algorithms of solution B differ in the calculated call dropping probability (see section 3.2) and do not stop and re-allocate the connections but instead immediately drop some of them in proportion to the detected overload. This blocking probability is the smallest with algorithm $B3$, and highest with $B1$. It can be seen that -as expected- the lower the call blocking ratio is, the higher the maintained utilization is. Depending on the measurement time base (S), the retained utilization is above, upon or below the target level. It can be seen that with different measurement time bases, different drop ratio approximations perform best while solution A is almost indifferent to the measurement time base (see Fig. 2/a and b).

Fig. 3 shows short term transients of the algorithms. Solution A is shown in Fig. 3/a for three different measurement time bases. It is interesting to see that since user traffic had a well defined period of 20 msec, measurements with 50 msec time base introduced high level of oscillation. It can be also seen that it takes a couple of measurement periods to bring the load back around the desired level though the control time is still in the order of 100 msec.

Solution B variants react very similarly (see Fig. 3/b) where $B1$ and $B2$ operations result in exactly the same transients.

Packet drop ratios: Here, only some representative results are presented due to the space limitations. As expected, the shorter the measurement time base is the shorter the congestion period will be (faster actions), hence packet drop periods decrease (see Fig. 4/a-b). The difference in operation between solution A and B can be seen when increasing the measurement windows. Since solution A stops user data flow its packet drop ratio more rapidly decreases with increasing measurement time base.

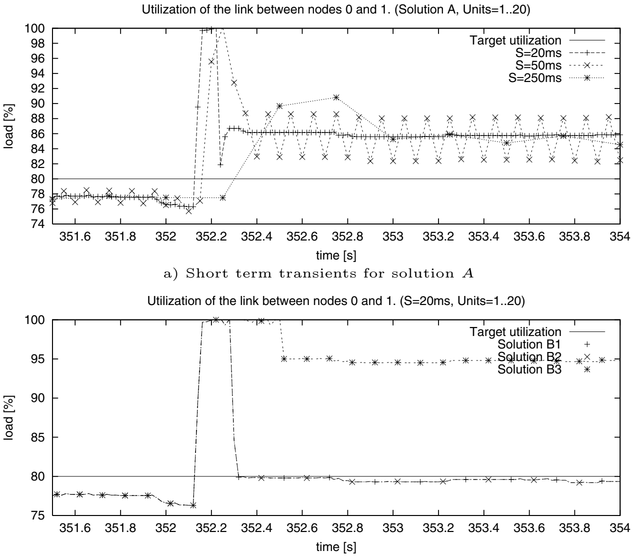
Signaling overheads: Fig. 4/c shows the protocol messages and the reservation status for solution A. It can be seen that after the detection of severe congestion all the 160% traffic load is released and tried to be reallocated (see the almost coinciding curves at 352 in Fig. 4/a). It can also be well seen that due to the synchronized re-initiation, refresh messages arrive in bursts. This



a) Utilization with $S = 20$ msec

b) Utilization with $S = 250$ msec

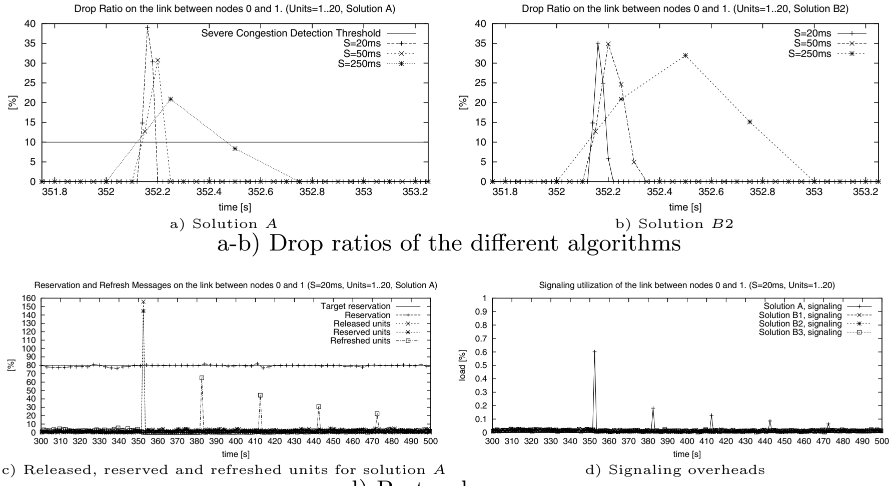
Fig. 2. Utilization



a) Short term transients for solution A

b) Short term transients for solution B descendants

Fig. 3. Short term transients



a) Solution *A*

b) Solution *B*2

a-b) Drop ratios of the different algorithms

c) Released, reserved and refreshed units for solution *A*

d) Signaling overheads

c-d) Protocol messages

Fig. 4. Protocol performances

characteristic will only fade out with several refresh times, as connections are terminating by nature. Fig. 4/d shows the signaling overhead for the various algorithms. It is obvious that for solution *B* descendants there are no increase in signaling overhead as shown in Fig. 4/d. On the other hand, due to the excess signaling introduced by solution *A* the overhead bristles appear. This however, is still quite negligible compared to the link capacity (see Fig. 4/d).

5 Conclusions

In this work we have shown some aspects of severe congestion handling with the RODA protocol [12]. We have designed and presented two basic algorithms that could cope with severe congestion situations in the order of network round trip times. This reaction time can be considered as close to optimal due to the transmission constraints imposed by the system. The presented algorithms differ in their transients but we can conclude that two of our solution *B* derivatives performed best in all situations with measurement time base equal to the framing time of the data traffic. In this very special case the two algorithms resulted in the same operation due to the traffic characteristics and differences only appeared with higher measurement time bases. Overall, we are aware of the need for further analysis in this area with more general traffic models (e.g. VBR); with multiple traffic classes (e.g. voice, video and best-effort); with more complex network topology (e.g. a concrete RAN topology) and with a comparsion to other resource management protocols (e.g. RSVP). Nevertheless, we believe that our current results can already be applied to certain special networks like RANs. We suppose that these results will trigger new dialogs from the community.

Acknowledgments. The authors owe special thanks to the *RMD research team* located in Sweden, Netherlands and Hungary. This work has partially been done at the High Speed Networks Laboratory, at the DTT of BUTE.

References

1. Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z., Weiss, W.: An architecture for differentiated service. Request for Comments 2475, Internet Engineering Task Force (1998)
2. Partain, D., Karagiannis, G., Wallentin, P., Westberg, L.: Resource reservation issues in cellular access networks. Internet Draft, Internet Engineering Task Force (2002) Work in progress.
3. Westberg, L., Jacobson, M., Karagiannis, G., Oosthoek, S., Partain, D., Rexhepi, V., Szabó, R., Wallentin, P.: Resource management in diffserv (RMD) framework. Internet-draft: draft-westberg-rmd-framework-xx.txt, Internet Engineering Task Force (2002) work in progress.
4. Elek, V., Karlsson, G., Ronngren, R.: Admission control based on end-to-end measurements. In: Proceedings of the Conference on Computer Communications (IEEE Infocom), Tel-Aviv, Israel (2000)
5. Breslau, L., Jamin, S., Shenker, S.: Comments on the performance of measurement-based admission control algorithms. In: Proceedings of the Conference on Computer Communications (IEEE Infocom), Tel Aviv, Israel (2000)
6. Braden, R., Ed., Zhang, L., Berson, S., Herzog, S., Jamin, S.: Resource ReSerVation protocol (RSVP) – version 1 functional specification. Request for Comments 2205, Internet Engineering Task Force (1997)
7. Feher, G., Nemeth, K., Maliosz, M., Cselenyi, I., Bergkvist, J., Ahlard, D., Engborg, T.: Boomerang - a simple protocol for resource reservation in ip networks. In: IEEE Workshop on QoS Support for Real-Time Internet Applications, Vancouver, Canada (1999)
8. Baker, F., Iturralde, C., Faucheur, F.L., Davie, B.: RSVP reservations aggregation. Internet Draft, Internet Engineering Task Force (2001) Work in progress.
9. Gibbens, R.J., Kelly, F.P.: Resource pricing and the evolution of congestion control. Automatica **35** (1999) 1969–1985
10. Heijenk, G., Karagiannis, G., Rexhepi, V., Westberg, L.: Diffserv resource management in ip-based radio access networks. In: Wireless Personal Multimedia Communications (WPMC'01), Aalborg, Denmark (2001)
11. Ádám Marquetant, Pop, O., Szabó, R., Dinnyés, G., Turányi, Z.: Novel enhancements to load control - a soft-state, lightweight admission control protocol. In: to appear at QofIS2001 - 2nd International Workshop on Quality of future Internet Services, Coimbra, Portugal, COST263 (2001)
12. Westberg, L., Jacobsson, M., Karagiannis, G., Oosthoek, S., Partain, D., Rexhepi, V., Wallentin, P.: Resource management in diffserv on demand (RODA) PHR. Internet Draft, Internet Engineering Task Force (2001) Work in progress.
13. The network simulator - ns-2. (http://www.isi.edu/nsnam/ns/)

Resource Allocation with Persistent and Transient Flows

Supratim Deb[1], Ayalvadi Ganesh[2], and Peter Key[2]

[1] Coordinated Science Lab., University of Illinois at Urbana-Champaign, 1308 W. Main Street, Urbana, IL 61801, USA deb@uiuc.edu

[2] Microsoft Research, 7 J J Thomson Ave., Cambridge CB3 0FB, UK
ajg,peterkey@microsoft.com

Abstract. The flow control algorithms currently used in the Internet have been tailored to share bandwidth between users on the basis of the physical characteristics of the network links they use rather than the characteristics of their applications. This can result in a perception of poor quality of service by some users even when adequate bandwidth is potentially available, and is the motivation for seeking to provide differentiated services. In this paper, stimulated by current discussion on Web mice and elephants, we explore service differentiation between persistent and short-lived flows, and between file transfers of different sizes. In particular, we seek to achieve this using decentralized algorithms that can be implemented by end-systems without requiring the support of a complex network architecture. The algorithms we propose correspond to a form of weighted processor sharing and can be tailored to approximate the shortest remaining processing time service discipline.

Keywords: Service differentiation, bandwidth allocation, decentralized control, weighted processor sharing, shortest remaining processing time.

1 Introduction

Most data in the current Internet is transferred using TCP. This protocol has two phases: a slow start phase which probes for available bandwidth up to a certain threshold, and a subsequent congestion avoidance phase that attempts to stabilize around a fair share. Despite having an aggressive ramp up phase, the throughput during slow start is typically much less than in the congestion avoidance mode due to the small size of the initial window, time-outs triggered by packet loss, etc. Moreover, the fair shares reached in the congestion avoidance phase allocate equal bandwidth to all file transfers having the same round-trip time and access bandwidth, irrespective of the sizes of the files being transferred. This results in a poor response time for short file transfers and raises the question of whether it is possible to improve performance for short file transfers without significantly degrading it for long file transfers. This question assumes particular importance in the context of the finding by a number of researchers that file sizes on the Web have a heavy-tailed distribution [6]: when file sizes vary over several

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 455–466, 2002.

orders of magnitude, treating all file transfers identically may not be appropriate. This has led to research on improving the throughput of short flows, either by altering the slow-start behavior [2,11,3], or by putting short flows into a different class [19,10], or by providing a predictive service to long flows [5].

A related problem is that of sharing bandwidth between file transfers and real-time traffic such as Internet telephony or video conferencing. Real-time flows are usually long-lived and can be treated as persistent sources for purposes of analysis. They have very different quality of service requirements from file transfers. Whereas what matters for file transfers is usually the transfer time, or equivalently, average bandwidth over the entire transfer period, real-time flows typically care about the bandwidth they receive at each instant in time (or, more precisely, averages over time periods much shorter than the lifetime of the connection). The value of bandwidth to a user can be described mathematically by a utility function which captures elements of the quality of service perceived by the user. Utility functions are commonly used in economics to represent individual preferences and to address questions of fair allocation. The resource allocation problem can be cast as one of maximizing the aggregate utility of all users.

We model the utility for a file transfer as the negative of the time taken to complete the transfer. For real-time traffic, we assume that the total utility obtained is the integral of an instantaneous utility over the lifetime of the connection; the instantaneous utility, in turn, is modeled as an increasing and concave function of the bandwidth received by the flow at that instant. Such a concave function reflects diminishing marginal utility to the user as the allocated bandwidth increases. Equivalently, concavity models a preference on the part of the user for a fixed constant bandwidth over a fluctuating bandwidth allocation with the same mean. Sources with such a utility are referred to as *elastic* sources in the literature. There has recently been considerable work on bandwidth sharing between persistent elastic users [14,17]. However, the problem of combining such sources with transient sessions such as file transfers has received little attention. One recent study [15] suggests that, when the two traffic types share a network, file transfers should receive priority.

Our main results and the organization of the paper are as follows. In Section 2, we consider persistent elastic sources sharing a link with transient sessions transferring a fixed volume of data. We pose the bandwidth allocation problem as an optimization problem and solve it numerically. We then derive practical flow-control schemes that can be easily implemented in a decentralized manner, and show that these are close to optimal. In Section 3, we consider a scenario where the transient sessions have different amounts of data to transfer. The shortest remaining processing time (SRPT) policy yields the optimal bandwidth allocation. We propose a practical scheme that approximates SRPT and study its performance through simulation. We show that there is an advantage to increasing the throughput given to short flows, and that this can be done without appreciably penalizing long flows. We present our conclusions and discuss directions for future research in Section 4.

2 Bandwidth Sharing between Persistent and Transient Flows

Consider a single link of capacity C shared by a fixed number, K, of persistent flows and a variable number of short-lived flows (also called Mice). The persistent flows are modeled in aggregate as having an increasing and strictly concave instantaneous utility function $KU_e(x_e)$, where x_e is the aggregate bandwidth allocated to these flows at a specified time. The utility over a time period is given by the integral of the instantaneous utility over that period. To simplify technicalities in the analysis below, we assume in addition that U_e is differentiable. Short flows correspond to file transfers. They arrive into the system at the points of a Poisson process of rate λ and leave when the file transfer is complete. The file sizes are assumed to be exponentially distributed with mean f. Let $\rho = \lambda f/C$ denote the load offered by the short flows. We shall assume that $\rho < 1$.

There is a unit holding cost per unit time for each short flow in the system. The goal is to maximize the time average of $KU_e(x_e(t)) - N(t)$, where $N(t)$ denotes the number of short flows in the system at time t. To this end, we introduce the performance objective,

$$J_\pi(n) = \lim_{T\to\infty} \frac{1}{T} E\left[\int_0^T [KU_e(x_e(t)) \ - \ N(t)]dt \ \middle|\ N(0) = n \right] \tag{1}$$

and seek a policy π that maximizes this objective. By Little's law, the time average of $N(t)$ is the same as λ times the mean sojourn time of a file transfer, so the objective is to maximize the utility of long flows, subject to a bound on the mean sojourn time of file transfers. The objective function above is precisely the Lagrangian for this optimization problem.

We seek stationary optimal policies for the optimization problem described above. By the assumption of exponential file sizes, the state of the system is fully described by the number of short flows in progress and we have a semi-Markov decision problem. If the number of short flows is restricted to some $n_{\max}$, and non-zero capacity is allocated to short flows whenever any are present, then the Markov process is irreducible and has a finite state space. Under these conditions, it can be shown that there is a stationary optimal policy, and that it can be computed numerically using value iteration. The proof is omitted due to lack of space, but can be found in [8] along with a discussion of structural properties of the optimal control policy.

In order to compare the optimal policy with sub-optimal policies that we shall consider below, we need the following elementary bound on the performance of the optimal policy.

Lemma 1. *Suppose the state space is not truncated, i.e., $n_{\max} = \infty$. Then, for any policy π and any initial state n, we have $J_\pi(n) \le KU_e((1-\rho)C)$.*

Proof. Since the load offered by the short flows is ρ, any policy π that allocates capacity less than ρC to these flows on average will be unstable in the sense that

$N(t) \to \infty$ as $t \to \infty$. Thus, for any such policy, $J_\pi(n) = -\infty$, starting from any n. Therefore, we can restrict attention to policies π that, on average, allocate capacity no more than $(1-\rho)C$ to the persistent flows. Since U_e was assumed to be concave, we now obtain from Jensen's inequality and the non-negativity of $N(t)$ that

$$\frac{1}{T}\int_0^T [KU_e(x_e(t)) - N(t)] \le KU_e\left(\frac{1}{T}\int_0^T x_e(t)\right) .$$

Taking expectations and using Jensen's inequality once more, we get

$$J_\pi(n) \le KU_e\left(E\left[\frac{1}{T}\int_0^T x_e(t)|N(0)=n\right]\right) \le KU_e((1-\rho)C) , \qquad (2)$$

since $E[\frac{1}{T}\int_0^T x_e(t)] \le (1-\rho)C$ for all T sufficiently large, and U_e is an increasing function. □

Implementing the optimal policy requires knowledge of the number of short flows in progress and may not be practical. This leads us to consider simpler policies that are practically realizable. We show for two such policies below that they are close to optimal. In the rest of the paper, we will work with utility functions of the form

$$U_e(x_e) = \frac{1}{1-\beta}\left(\frac{x_e}{C}\right)^{1-\beta} , \quad \beta > 0. \qquad (3)$$

If $\beta = 1$, we take $U_e(x_e) = \log(x_e/C)$. These constitute a fairly general class of utility functions and have been considered by a number of authors; see, for example, [18]. The bandwidth shares assigned by TCP approximately maximize a utility function of this form with $\beta = 2$.

Static Policy. A fixed amount of bandwidth $\tilde{C} < C$ is reserved for the persistent sources and the remainder is shared equally among file transfers. This can be implemented by logically partitioning the link between persistent and short flows and using TCP for the short flows, for example.

Now, irrespective of the file size distribution, the number of short flows in progress evolves like the queue size in an $M/G/1-PS$ queue, with load $\alpha = \lambda f/(C-\tilde{C})$. The equilibrium queue length distribution is geometric with parameter α (see [13], for example), and so the mean number of short flows in progress is $E_\pi[n] = \alpha/(1-\alpha)$. The bandwidth allocated to persistent flows, $\tilde{C}$, can be expressed as $\tilde{C} = C - (\lambda f/\alpha) = (\alpha-\rho)C/\alpha$. Hence,

$$E_\pi[KU_e(x_e(n)) - n] = KU_e\Big(\frac{\alpha-\rho}{\alpha}C\Big) - \frac{\alpha}{1-\alpha} . \qquad (4)$$

Taking $\alpha = 1 - a/\sqrt{K}$ and using (3), we obtain

$$E_\pi[KU_e(x_e(n)) - n] = \frac{a\sqrt{K}}{1-\beta} + \frac{K - a\sqrt{K}}{1-\beta}\left(\frac{1-\rho-(a/\sqrt{K})}{1-(a/\sqrt{K})}\right)^{1-\beta} - \frac{\sqrt{K}}{a} + 1$$

$$= K\frac{(1-\rho)^{1-\beta}}{1-\beta} + O(\sqrt{K}) \;=\; KU_e((1-\rho)C) + O(\sqrt{K}) \; . \tag{5}$$

Recall that ρC is the rate at which work is brought in by short flows, and αC is the capacity allocated to them. The choice $\alpha = 1 - a/\sqrt{K}$ corresponds to allocating most of the available capacity to short flows, reserving only a small fraction $a/\sqrt{K}$ for persistent sources.

How much worse than optimal is the static policy? One way to quantify this is to ask how large a capacity $\hat{C}$ is needed, so that the total utility achieved using the static policy on a link of capacity C is the same as the utility achieved using the optimal policy on a link of capacity $\hat{C}$. Recall that, by (2),

$$E_\pi[KU_e(x_e(n)) - n] \le \frac{K}{1-\beta}\left[\frac{(1-\rho)\hat{C}}{C}\right]^{1-\beta}$$

for a link of capacity $\hat{C}$, using any policy. Comparing this with (5), we see that $\hat{C} = C(1 - O(1/\sqrt{K}))$. In so far as K is large in the typical operating regime of interest, this shows that the static policy is close to optimal.

Implementation of the static policy requires that bandwidth partitioning be carried out by network routers. In contrast, the weighted processor sharing policy discussed next can be implemented at end systems.

Weighted Processor Sharing. Suppose each persistent source has weight 1 and each file transfer in progress has weight w, and that capacity is shared between users in proportion to their weights. In particular, each file transfer in progress gets the same share of capacity. Thus, irrespective of the file size distribution, the number of file transfers in progress can be modeled by a symmetric queue (see Lemma 3.9 in Kelly [13]), and has the invariant distribution of a birth-death process with constant birth rate λ, and state-dependent death rate $\mu_n = (C - x_e(n))/f$. Here $x_e(n)$ is the capacity allocated to persistent sources when n short flows are in progress, and f is the mean file size. If we assume further that $k := K/w$ is an integer, then it can be shown that the invariant distribution is given by

$$\pi(n) = \binom{k+n}{n}\rho^n(1-\rho)^{k+1} \; , \tag{6}$$

and a simple calculation yields $E_\pi[n] = (k+1)\rho/(1-\rho)$ for the mean number of short flows in the system. Details are omitted for brevity, but can be found in [8]. It is not possible to obtain a closed-form expression for $E_\pi[U_e(x_e(n))]$ in general, but we can obtain approximations using a Taylor expansion for U_e when k is large. After routine calculations detailed in [8], we obtain

$$E_\pi[KU_e(x_e(n)) - n] \approx \\ \frac{K(1-\rho)^{1-\beta}}{1-\beta} + \frac{K\rho(1-\rho)^{1-\beta}}{k} - \frac{K\beta\rho(1-\rho)^{1-\beta}}{2k} - \frac{(k+1)\rho}{1-\rho} \; . \tag{7}$$

Recall that $k = K/w$, where w is the weight given to short-lived flows. It follows from the above that, by choosing $w = \sqrt{K}$, we get $E_\pi[KU_e(x_e(n)) - n] \geq KU_e((1-\rho)C) - O(\sqrt{K})$. Since no policy can achieve a total utility greater than $KU_e((1-\rho)C)$, we conclude that the minimum bandwidth, $\hat{C}$, required by an optimal policy to achieve the same utility as achieved by the weighted processor sharing policy is given by $\hat{C} = C(1 - O(1/\sqrt{K}))$.

Thus, the weighted processor sharing policy is nearly optimal, in the same sense as the static policy. Moreover, it can be implemented by the end systems rather than the network, for example by having end systems use a weighted analogue of TCP with weights chosen as above. An alternative implementation would be to use a willingness-to-pay scheme, as described in [9], with a willingness-to-pay parameter proportional to the weights above.

Numerical Results. We now derive explicit formulas in the special case $\beta = 2$, i.e., $U_e(x) = -C/x$. Recall that this is the utility function implicitly maximized by TCP. The static policy allocates fixed capacity $\rho C/\alpha$ to the short flows; when $\beta = 2$, we obtain from (4) that the optimal value of α is $\alpha = 1 - \frac{1-\rho}{1+\sqrt{K\rho}}$, and that

$$E_\pi[KU_e(x_e(n))] \;=\; -\,\frac{K + \sqrt{K\rho}}{1-\rho}\;, \quad E_\pi[n] \;=\; \frac{\sqrt{K\rho} + \rho}{1-\rho}\;.$$

When $\beta = 2$, we can also explicitly calculate $E_\pi[U_e(x_e(n))]$ for the weighted-PS policy. We obtain

$$E_\pi[U_e(x_e(n))] = E_\pi\left[U_e\left(\frac{k}{k+n}C\right)\right] = -E_\pi\left[\frac{k+n}{k}\right] = -\frac{k+\rho}{k(1-\rho)}\;.$$

A simple calculation now yields that the optimal value of k is $\sqrt{K}$, i.e., each transient flow should be given a weight $w = \sqrt{K}$ relative to each persistent flow. With this choice of k, we get

$$E_\pi[KU_e(x_e(n))] \;=\; -\,\frac{K + \sqrt{K}\rho}{1-\rho}, \quad E_\pi[n] \;=\; \frac{\sqrt{K}\rho + \rho}{1-\rho}.$$

We compare the mean utility and number in system for the static and weighted-PS policies with those for the optimal policy, obtained numerically. For this purpose, we choose the system parameters $C = 1000$, $K = 25$, $f = 100$, and vary λ so that $\rho = \lambda f/C$ spans the interval $[0.1, 0.7]$. We truncate the state space at $n_{\max} = 100$ for the value iterations. The results are plotted below. Figure 1 shows the mean utility for the optimal, static and weighted policies, while Figure 2 shows the mean number of short flows in progress for each policy. The figures show that neither the persistent nor the transient flows suffer much by using the sub-optimal policies considered. Figure 3 shows the additional capacity required by the static policy if it is to achieve the same total utility as the optimal policy; 3(a) corresponds to $K = 25$ and 3(b) to $K = 5$. We see that the loss incurred by the sub-optimal policies is small, even for small values of K.

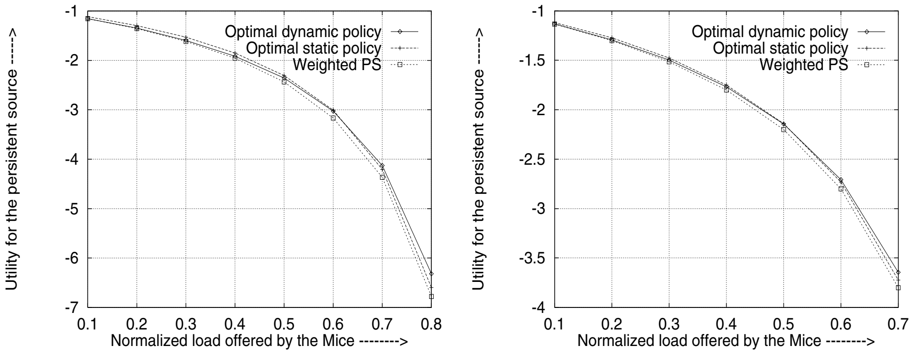


Fig. 1. Average utility of the long flow under three different allocation strategies. $C = 1000$, Mean file size=100, $U_e(x) = \frac{-C}{x}$, $n_{max} = 100$. $K = 5$ in the *left panel* and $K = 25$ in the *right panel*. The arrival rate is varied along the x-axis

3 Bandwidth Sharing between Transient Flows

We now consider how capacity should be shared between file transfers when the sizes of the files being transferred might vary over several orders of magnitude. If the objective is to minimize the number of file transfers in progress (equivalently, the mean holding cost or mean sojourn time) and the amount remaining to be transferred is known, then a simple interchange argument shows that the optimal policy is to give priority to the file with shortest remaining processing time (SRPT). This policy has been proposed in the context of Web servers [12, 1]. However, it is not suited to our problem for a couple of reasons. First, it needs a centralized controller to assign priority (or a distributed leader election protocol, which imposes a high overhead). Second, while the concept is clear for a single bottleneck link or resource, it does not generalize easily to multiple bottlenecks. This motivates us to consider a generalization of the weighted PS policy introduced in the previous section and show that it can be made to approximate SRPT. Though the analysis and simulations in this paper pertain to a single link, the algorithms we propose generalize easily to networks.

We also note that, in networks, the stability region of priority policies such as SRPT is not easily obtained; it is known that the "$\rho < 1$" condition that the offered load on each link be smaller than its capacity is not sufficient for stability. On the other hand, this condition does guarantee stability for the algorithms we consider, as shown in [4]. That is another advantage of the proposed algorithms in the network context.

We continue to work with the optimization problem posed in the previous section. There, we considered how to split capacity between persistent and transient flows but did not consider further how the capacity allocated to transient flows should be shared between them. If file sizes are exponentially distributed and the allocation decision has to be made without knowing the sizes of all file transfers in progress, then it does not matter how this allocation is made; any

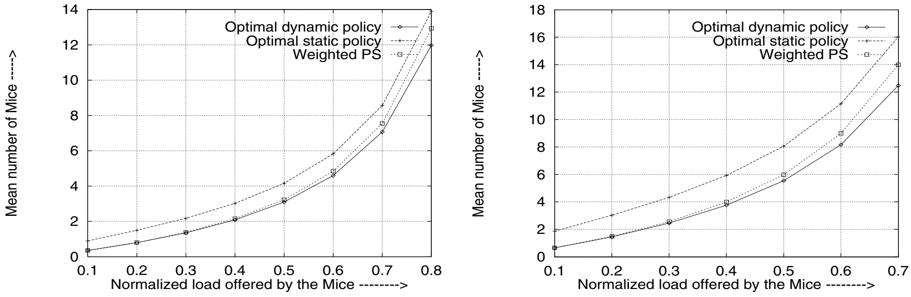


Fig. 2. Average number of short flow under three different allocation strategies. $C = 1000$, Mean file size=100, $U_e(x) = \frac{-C}{x}$, $n_{max} = 100$. $K = 5$ in the *left panel* and $K = 25$ in the *right panel*. The arrival rate is varied along the x-axis

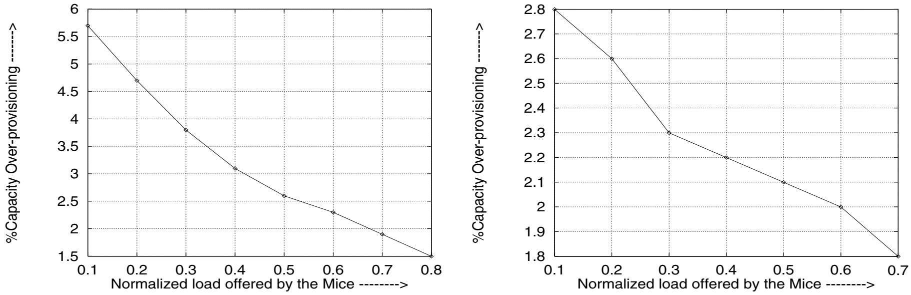


Fig. 3. Capacity over-provisioning sufficient for optimal static allocation to outperform optimal dynamic allocation. $C = 1000$, $f = 100$, $U_e(x) = \frac{-C}{x}$, $n_{max} = 100$. $K = 5$ in the *left panel* and $K = 25$ in the *right panel*

allocation that doesn't leave capacity idle achieves the same mean number in system. If file sizes are known or if they aren't exponentially distributed, then this is no longer true; for example, if file sizes are heavy-tailed, the first-come-first-served policy performs worse than processor-sharing. We noted above that, if file sizes are known, then SRPT is optimal.

We now consider a weighted processor sharing policy where each transient flow chooses its own weight based on its residual file size. Suppose the weights are chosen according to

$$w_i = w_{min} \ + \ (w_{max} - w_{min}) \exp(-af_i^r) \ , \tag{8}$$

where w_i and f_i^r denote the weight assigned to the i^{th} flow and its residual file size, and w_{min}, w_{max} and a are system parameters. The link capacity C is shared between flows in proportion to their weights, i.e., flow i receives capacity w_iC/W, where W denotes the sum of w_i over all flows in the system. A similar policy has been proposed recently in [7].

We shall assume that W is constant over time. Such an assumption is plausible in a large system operating in a steady-state regime. In particular, if the system carries a large number of persistent flows, then the fluctuation in W is only due to short flows entering and leaving the system, and can be neglected to a first approximation. With this assumption, we can calculate the sojourn time of a file transfer as a function of the initial file size. Letting f_i denote the size of file i, we have

$$f_i^r(0) = f_i \ , \quad \frac{d}{dt} f_i^r(t) \ = \ - \ \frac{w_i(t)}{W} C \ , \tag{9}$$

where $w_i(t)$ is specified in terms of $f_i^r(t)$ via (8). Here, t denotes the time since the arrival of flow i into the system. Let $T_i = \inf\{t > 0 : f_i^r(t) = 0\}$ denote the sojourn time of flow i. A straightforward calculation using (8) and (9) yields

$$T_i = \frac{W}{aCw_{min}} \log \left[1 + \frac{w_{min}}{w_{max}} \left(e^{af_i} - 1 \right) \right] \ .$$

The (unweighted) processor sharing policy is recovered in the limit $a \to 0$, in which case $T_i = Wf_i/w_{max}$. The sojourn time of a file is thus proportional to its size, which is desirable in terms of fairness but has the disadvantage that small files see poor performance.

In order to quantify the extent to which the proposed service discipline favors short flows, we compute the ratio of sojourn times for two different files, of sizes f_1 and f_2. With plain sharing, this ratio is $T(f_1)/T(f_2) = f_1/f_2$. Denoting the ratio w_{min}/w_{max} by α, we obtain for the scheme proposed above that

$$\frac{T(f_1)}{T(f_2)} = \frac{\log \left[1 + \alpha(e^{af_1} - 1) \right]}{\log \left[1 + \alpha(e^{af_2} - 1) \right]} \ . \tag{10}$$

We observe that if f_1 and f_2 are both large relative to $1/a$ and if, moreover, αe^{af_i} is much bigger than 1 for $i = 1, 2$, then $T(f_1)/T(f_2) \approx f_1/f_2$. In other words, the *stretch*, defined as the ratio of sojourn time to file size, is roughly constant for large files, meaning that the scheme approximates processor sharing at large file sizes. In particular, it avoids starvation of very large file transfers. On the other hand, if f_1 and f_2 are both small relative to $1/a$, then again $T(f_1)/T(f_2) \approx f_1/f_2$. Finally, suppose f_1 is large and f_2 is small relative to $1/a$. Then, by (10),

$$\frac{T(f_1)}{T(f_2)} \approx \frac{af_1 + \log \alpha}{\alpha a f_2} \approx \frac{1}{\alpha} \frac{f_1}{f_2} = \frac{w_{max}}{w_{min}} \frac{f_1}{f_2} \ .$$

In other words, the large file has a stretch approximately $1/\alpha$ times greater, or receives a bandwidth share approximately $\alpha = w_{min}/w_{max}$ as much as a small file. Loosely speaking, files much smaller than $1/a$ are “mice”, files much larger than $1/a$ are “elephants”, all mice are treated roughly equally, as are all elephants, but mice are favored over elephants. Note that this is achieved without explicitly splitting files into classes, but simply by having them choose individual weights based on their residual file sizes.

The degree to which mice are favored is determined by the ratio $1/\alpha = w_{max}/w_{min}$. This can be seen clearly in Figure 4, where we have plotted the stretch, $T(f)/f$, as a function of the normalized file size af over the range $[0, 20]$. We take $W/(Cw_{max}) = 1$ for convenience. From top to bottom, the 3 curves on the plot correspond to $1/\alpha = 5, 10$ and 20 respectively.

The plots suggest that large files receive much less capacity on average than do short files. It needs to be kept in mind that this is under the assumption that W is constant, which is not valid if there are no persistent flows. A model with no persistent flows and with an SRPT service discipline has been studied in [1], where it is shown that the stretch of long flows remains bounded. The intuition is that there will be epochs when the long flow is competing with very few or no short flows, at which times it is not handicapped by its small weight. A similar intuition applies to our model, and in fact the plots of stretch in Figure 4 correspond to "worst-case" values.

Simulation Results. We simulate a system with capacity $C = 1000$ carrying $K = 25$ persistent flows, each of which has weight 1 and has the utility function $U_e(x) = -C/x$. File transfers arrive at rate λ, and file sizes have the Pareto distribution, $P(\text{file-size} > x) = 1/(1 + (x/f))^2$, $x \geq 0$, with mean file size $f = 100$. We take $a = 1/f$, $w_{max} = 50$ and $w_{min} = 10$. Performance measures for processor sharing with the scheme described above, and processor sharing with constant (file-size independent) weights $\overline{w}$ for three different weights, 10, 50 and 25, are shown in Figure 5. The left panel shows the utility received by the persistent flows under each policy. The panel on the right shows the mean number of transient flows in the system. The simulation results are based on 12,000 events (file arrivals) with a burn-in period of 1000 time units for the system to reach stationarity, and, are averaged over multiple runs. Clearly, when $\overline{w} = 50$, the average stretch of the transient flows go down but at the cost of a reduced utility for the persistent flow. When $\overline{w} = 10$, the persistent flows perform better but the average stretch of the short flows increases a lot. However, by using the processor sharing described in this section when the weights of the transient flows are varied in a dynamic manner, the transient flows can achieve a small stretch without starving the persistent flow much. We have also shown the plots for the case when the weights are kept constant at $\overline{w} = 25$. The proposed processor sharing scheme still does better.

4 Concluding Remarks

We considered the problem of optimal bandwidth allocation in a system consisting of both persistent and transient flows. Treating all transient flows as identical, we first described simple algorithms that achieve a nearly optimal partitioning of the available bandwidth between the persistent and transient sources. We then studied the problem of how to share the bandwidth allocated to transient flows among file transfers of different sizes. We described a distributed scheme

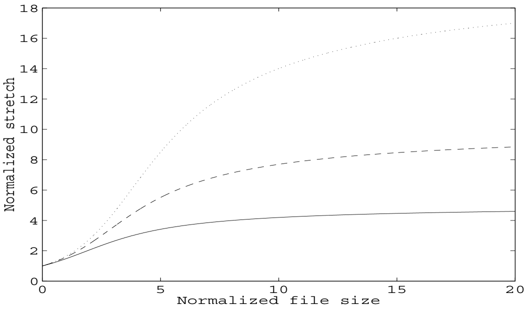


Fig. 4. Stretch vs. file size for $w_{max}/w_{min} = 20(top)$, $10(middle)$ and $5(bottom)$

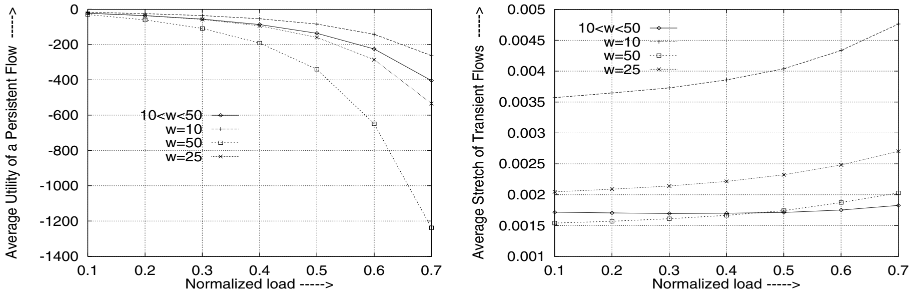


Fig. 5. Plots showing comparison of of average utility of a persistent flow (*left*), and, average stretch of short flows (*right*) with four schemes: in one the weights of the short flows are varied between 10 (w_{min}) and 50 (w_{max}) according to the scheme discussed in this section, and the other three are with fixed weights ($\overline{w}$) as 10, 50 and 25 respectively. The different parameters are, $C = 1000$, $f = 100$, $U_e(x) = \frac{-C}{x}, K = 25$ (the number of persistent flows). The arrival rate λ is varied along the x-axis

that can be viewed as approximating SRPT or, equivalently, as discriminating in favor of "mice" over "elephants".

The analysis in this paper pertains to idealized systems in which bandwidth is shared perfectly between users in proportion to their weights. In fact, such an allocation can be approximately achieved by decentralized adaptive mechanisms described in [14,9,16] etc., which can be implemented by end systems with minimal support from the network. Second, the optimal choice of parameters for the policies described in Section 2 requires knowledge of system parameters, which is often unrealistic. We believe that comparable performance can be achieved by adaptive policies that tune their parameters based on measurements, but this remains a topic for future research.

The policies studied here for a single link can be extended easily to networks. The extensions have the desirable property that the network is stable under the natural condition that the offered load at each resource is smaller than its capacity. This is in contrast to priority schemes which can be unstable even when this condition is satisfied. A detailed investigation of the performance of the algorithms described here in a network context is a subject for future work.

References

1. N. Bansal and M. Harchol-Balter, “Analysis of SRPT scheduling: investigating unfairness”, *Proc. ACM Sigmetrics*, 2001.
2. C. Barakat and E. Altman, “Performance of Short TCP Transfers”, *Proc. Networking*, Paris, 2000.
3. Y. Bhumralkar, J. Lung and P. Varaiya, “Network Adaptive TCP Slow Start”, 2000. http://www.path.berkeley.edu/∼varaiya/papers_ps.dir/jeng.pdf
4. T. Bonald and L. Massoulié “Impact of fairness on Internet performance”, *Proc. ACM Sigmetrics*, 2001.
5. D. D. Clark and W. Fang, “Explicit Allocation of Best-Effort Packet Delivery Service”, *IEEE/ACM Trans. Networking*, 6(4): 362–373, 1998.
6. M. E. Crovella and A. Bestavros, “Self-similarity in World Wide Web traffic: Evidence and possible causes”, *IEEE/ACM Trans. Networking*, 5(6): 835–846, 1997.
7. G. de Veciana, “Enhancing Both Network and User Performance Metrics for Networks Supporting Best Effort Traffic”, *Thirty-Ninth Annual Allerton Conference on Communication, Control, and Computing*, Allerton House, Illinois, 2001.
8. S. Deb, A. Ganesh and P. Key, “Resource allocation with persistent and transient flows”, Microsoft Research Technical Report, 2001. http://research.microsoft.com/scripts/pubs/view.asp?TR_ID=MSR–TR–2001–114
9. R. J. Gibbens and F. P. Kelly, “Resource pricing and the evolution of congestion control”, *Automatica*, 35: 1969–1985, 1999.
10. L. Guo and I. Matta, “The war between mice and elephants”, Technical Report, Boston University, BU-CS-2001-0005, 2001.
11. T. Hammann and J. Walrand “A new fair algorithm for ECN-capable TCP”, *Proc. Infocom*, 2000.
12. M. Harchol-Balter, M. Crovella and S. Park, “The case for SRPT scheduling in Web servers”, Technical Report, MIT-LCS-TR-767, 1998.
13. F. P. Kelly, *Reversibility and Stochastic Networks*, John Wiley and Sons, New York, 1979.
14. F. P. Kelly, A. Maulloo and D. Tan, “Rate control in communication networks: shadow prices, proportional fairness and stability”, *J. Oper. Res. Soc.*, 49: 237–252, 1998.
15. P. Key and L. Massoulié “User policies in a network implementing congestion pricing”, Workshop on Internet Service Quality Economics (ISQE), 1999.
16. R. J. La and V. Anantharam, “Charge-sensitive TCP and rate control in the Internet”, *Proc. Infocom*, 2000.
17. S. H. Low and D.E. Lapsley, “Optimization flow control – I: Basic algorithm and convergence”, *IEEE/ACM Transactions on Networking*, 7: 861–875, 1999.
18. J. Mo and J. Walrand, “Fair end-to-end window-based congestion control”, *IEEE/ACM Trans. Networking*, 8(5): 556–567, 2000.
19. S. Yilmaz and I. Matta, “On class based isolation of UDP, short lived and long lived flows”, *Proc. Ninth Intl. Symp. Modeling, Analysis And Simulation of Computer And Telecommunication Systems,* Cincinnati, 2001.

A Novel and Simple MAC Protocol for High Speed Passive Optical LANs

Chuan Heng Foh and Moshe Zukerman

ARC Special Research Center for Ultra-Broadband Information Networks
EEE Department, The University of Melbourne
Parkville, Vic. 3010, Australia
{chuanhf,m.zukerman}@ee.mu.oz.au

Abstract. In this paper, we propose a new MAC protocol for Gigabit Local Area Networks, called the Request Contention Multiple Access (RCMA) protocol. RCMA is proposed to operate in the 10BASE-FP Ethernet network topology at a gigabit data rate. It does not require the sophisticated WDM technology. Unlike the current IEEE 802.3z Gigabit Ethernet MAC protocol, RCMA is efficient and stable for a wide range of user numbers. Furthermore, it can support service differentiation with no additional overhead. Its performance under the saturation condition is analyzed and compared with performance of the current IEEE 802.3z Gigabit Ethernet MAC protocol, and significant performance advantage is demonstrated for RCMA.

1 Introduction

The challenge in developing Medium Access Control (MAC) protocols in Gigabit local area networks (LANs) is not only to achieve a simple as well as efficient protocol, but also to ensure that its efficiency is not affected as the number of shared users increases. This paper proposes a new MAC protocol for Gigabit LANs that overcomes the drawbacks of it predecessors in the Gigabit LAN environment and achieves efficient scheduling with minimum overhead and complexity.

MAC protocols can be classified to collision based, reservation based, and collision/reservation hybrid. Collision based protocols are simple but inefficient, reservation protocols are efficient but relatively complex. Since the introduction of the carrier sense multiple access with collision detection (CSMA/CD) protocol a quarter of a century ago, efforts have been made to develop protocols that are both efficient and simple by using both collision and reservation schemes. CSMA/CD has been considered one of the first MAC protocols that in some sense is a collision based as well as a reservation based protocol (see page 317 in [1] which views "the first portion of a packet as making a reservation for the rest"). However, interestingly, in Gigabit networks, where the transmission time of data frames become "small" relative to the propagation delay, CSMA/CD looses its reservation "affiliation" and it becomes a pure collision protocol in some cases.

There have been many protocols, such as IEEE 802.14 or DOCSIS, that use intelligent Central Controller (CC) to receive requests for bandwidth from multiplicity of stations. These requests are transmitted to the CC using contention minislots. In other words, these requests may collide and then retransmitted. After receiving the requests, the CC transmits scheduling information to the stations, which then transmit

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 467-478, 2002.

their data frame collision free. Other reservation protocols are based on the distributed control principle. Examples are the IEEE 802.5 token ring and the IEEE 802.6 Distributed Queue Dual Bus (DQDB). These protocols achieve collision free transmission at the cost of complex transceivers.

The CSMA/CD protocol has been retained by the IEEE 802.3z working group as the MAC protocol for access arbitration in shared Gigabit Ethernet [2]. Due to the high data rate, to achieve backward compatibility and guarantee the proper operation of CSMA/CD, the IEEE 802.3z working group introduced *carrier extension* operation. If a data frame is too short for collision detection purposes, senders must append predefined carrier signals to the short data frame for a period of time that is long enough for collision detect. Another modification to the protocol is the slot time parameter. It is increased by almost 10 times from 512-bit time in 10 or 100 Mb/s Ethernet to 4096-bit time. Consequently, each collision in Gigabit Ethernet results in a loss of 10 times more data than in the 10 or 100Mb/s Ethernet.

In this paper, we propose a new MAC protocol for Gigabit LANs. We call it the *Request Contention Multiple Access* MAC protocol (RCMA). RCMA has the following traits: (i) it is simple and based on distributed control principle, synchronization between stations is not required; (ii) it achieves efficient scheduling and fairness with minimum overhead, intelligence and complexity; (iii) it is more efficient than IEEE 802.3z, and unlike IEEE 802.3z that suffers from efficiency degradation as the number of stations increases, the performance of RCMA remains stable; and (iv) RCMA can easily accommodate service differentiation.

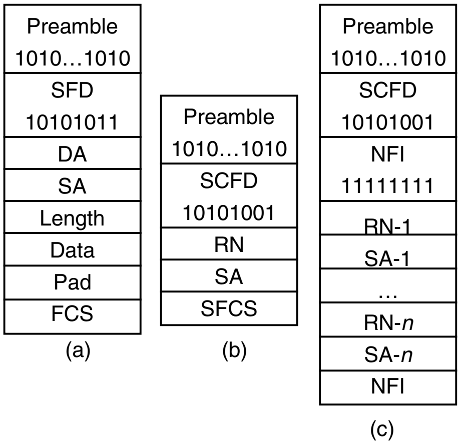
In RCMA, we propose that a station wishes to access the medium, if the medium is free, will first broadcast a very short request by which it will make a reservation for further data transmission. More importantly, the channel assignment task in RCMA will be performed in a distributive manner without the need for an intelligent CC. Because the RCMA request is much shorter than an IEEE 802.3 frame, the probability of collisions is significantly reduced. A new operation called non-contention channel assignment operation is introduced to exploit the short requests for further performance improvement in RCMA. In addition, service differentiation can be achieved by prioritizing the request. For some low priority services, stations can request for the channel access right with a lower request priority number so that delay sensitive services can be served first.

The paper is organized as follows. In Section II, we describe our proposed MAC protocol, RCMA, in detail. The performance analysis of RCMA is given in Section III. In Section IV, we compare RCMA with the IEEE 802.3z MAC protocol.

2 The RCMA Protocol

2.1 Network Topology

Our RCMA protocol is proposed to operate in a tree topology with a passive optical repeater similar to the 10BASE-FP Ethernet [3]. The data rate is expected to be 1Gb/s. The main advantage of this configuration is its cost effectiveness due to the use of passive optical repeaters.



Names of some important fields:
SA: Source Address (6 bytes)
RN: Request Number (1 byte)
NFI: NEXT Frame Indicator (1 byte)
SFD: Start Frame Delimiter (1 byte)
FCS: Frame Check Sequence (4 bytes)
SFCS: Short Frame Check Sequence (1 byte)
SCFD: Start Control Frame Delimiter (1 byte)

Fig. 1. The frame structure. (a) IEEE 802.3 frame; (b) RCMA Request Frame; (c) RCMA NEXT Frame

The main difference between the 10BASE-FP Ethernet and other Ethernet variations is the signal repeating mechanism. In 10BASE-FP, two optical fibers are connected to the passive optical repeater from each station, one for incoming, and another for outgoing traffic. When optical signals arrive at one port of the passive repeater, the signals will be repeated to all stations, including the originated station. Since the sender will receive its own transmission during its data frame transmission, collision detection is somehow difficult for the CSMA/CD protocol. Therefore, in Ethernet, a special transceiver is designed to allow the sender to detect collisions in the presence of its return signals.

However, in RCMA, a station is required to request access right before its actual data frame transmission can take place. Once the channel is reserved, the data frame is transmitted free of collision, thus collision detection operation is not required.

2.2 The RCMA Protocol

The key idea of our proposed protocol is that it makes use of the return signals repeated by a passive optical repeater to allow each sender to verify that its earlier transmission was successful. To make the operation more efficient, we introduce the use of a very small request frame for each sender contending for the channel access right to reserve the channel for longer data frame transmissions.

Let τ be the maximum signal propagation delay between any pair of the stations. When a station is ready for a data transmission, henceforth called a *ready station*, it is required to perform a *request contention* operation. It first prepares a request frame. The proposed request frame structure is depicted in Fig. 1(b). The station must randomly generate a 6-bit *request number* and store it in the *Request Number* (RN) field of the request frame. Its MAC address is also included. The request frame ends with an 8-bit *short frame check sequence* (SFCS) for error detection.

Before the request frame transmission, the station activates a timer called the *request-waiting timer* (RWTimer). RWTimer=$w \cdot T_s$, where w is a uniformly distributed random integer between zero and k-1, and T_s is the minislot time which is the time required to transmit the entire request frame plus a short guard time. The station waits and monitors the incoming channel after activated RWTimer. Detection of a request or data frame from the incoming channel generated by another station during that period will cause the station to abort its request frame transmission. This will ensure that the station which has requested the channel earlier has the priority to transmit based on first come first served principle.

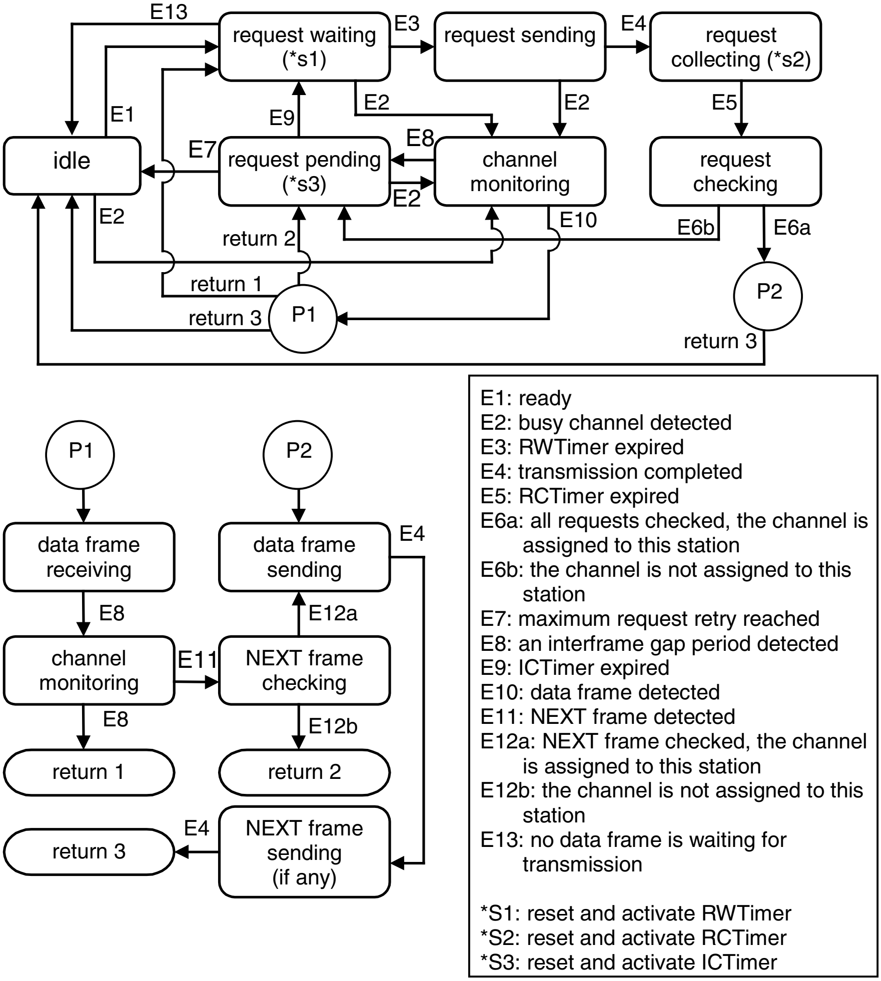
When RWTimer expires, if the incoming channel remains idle, the station may transmit its request frame. The station is required to monitor the incoming channel during its request frame transmission. If a carrier is detected on the incoming channel, the request frame transmission must also be aborted immediately. We assume that the cable between a station and the repeater is long enough (in the case of 1Gb/s and 16 bytes request frame, the cable must be at least 19.2m) so that the request frame will not return back to the originated station while transmitting that request frame. However, the request frame transmissions are subject to collisions. If two stations transmit the request almost at the same time such that the request frames meet at the passive repeater, these request frames are corrupted due to the overlapping signals. Otherwise, the request frames are considered successfully transmitted and can be read correctly by all stations.

After the very last bit of the request frame is transmitted, the station activates another timer called the *request-collection timer* (RCTimer). This timer is set to 2τ plus a short guard time. During this time interval, the station monitors the incoming channel and collects any request frame including its own request frame transmitted earlier. Any incorrect or incomplete request frames are discarded. When RCTimer expires, the station can be sure that all transmitted request frames have arrived and no further request frame is still propagating in the network. The station then compares all the collected requests. One of these requests will be the *winning request*. The winning request is the one with the largest request number among all collected requests. The station that originally sent the winning request will gain the channel access right. This station will henceforth be called the *winner*.

All stations contending for the channel access right, including the winner can identify the winner by comparing the request numbers, and identifying the MAC address of the largest request number. No collision detection is required during the data frame transmission. The choice of data frame structure is optional; here we propose to use the IEEE 802.3 frame [3] shown in Fig. 1(a).

It is possible that two or more stations may choose the same request number. There are many ways to break this tie. Under one simple option, the station of a larger numerical MAC address always has the advantage to transmit its data frame first. This will not affect the fairness significantly because this event is very rare.

Since the station, which has given the exclusive right to access the channel, is also aware of other requests while competing for the channel access right, after its data frame transmission and an *interframe gap* (IFG) period similar to its Gigabit Ethernet counterpart, it may transmit a special control frame, called the NEXT frame. The proposed frame structure for the NEXT frame is given in Fig. 1(c). The NEXT frame generally contains a list of successful requests that the station collected earlier, sorted by request number.



E1: ready
E2: busy channel detected
E3: RWTimer expired
E4: transmission completed
E5: RCTimer expired
E6a: all requests checked, the channel is assigned to this station
E6b: the channel is not assigned to this station
E7: maximum request retry reached
E8: an interframe gap period detected
E9: ICTimer expired
E10: data frame detected
E11: NEXT frame detected
E12a: NEXT frame checked, the channel is assigned to this station
E12b: the channel is not assigned to this station
E13: no data frame is waiting for transmission

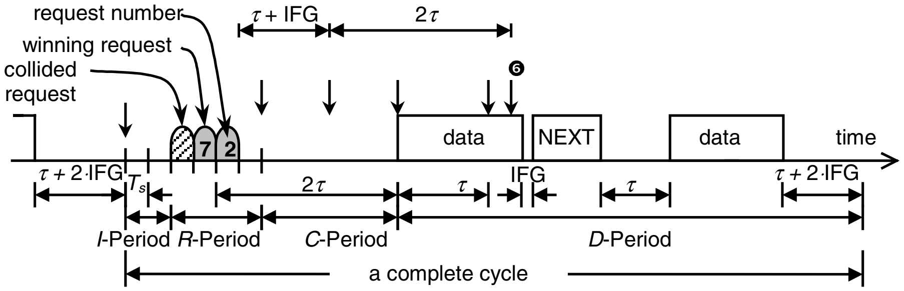
*S1: reset and activate RWTimer
*S2: reset and activate RCTimer
*S3: reset and activate ICTimer

Fig. 2. The finite state machine of a RCMA transceiver

When the NEXT frame is transmitted, RCMA enters a *non-contention channel assignment* operation. Each station, after receiving the NEXT frame, compares its MAC address with the first MAC address in the NEXT frame. If matched, that station may transmit its data frame after an IFG period. Again, the station removes its record from the NEXT frame and transmits the modified NEXT frame after it has completed its data frame transmission.

When the last station on the list in the NEXT frame completes its data frame transmission, no NEXT frame will follow. After an IFG period, all ready stations enter the request contention operation.

If all request frames collide, no data frame transmission will occur. After discovering that there is no data frame transmission, all ready stations immediately repeat the request contention operation to compete for the channel access right.



all stations detects the end of the data frame transmission;

all stations detects the first request frame transmission, the stations that have not transmitted their request frames must abort their request frame transmissions;

the channel turns from busy to idle due to request frame of request number '2'. After detecting the channel to be idle for an IFG period, all stations, that do not participate in request contention, reset and activate their ICTimers which will expire at ❻;

the RCTimer of the winner expires, it starts its data frame transmission immediately;

all stations sense the busy incoming channel and will not access the channel at ❻;

Fig. 3. The snapshot of RCMA channel

A critical aspect of RCMA is its implicit channel assignment property. If the assigned station fail to initiate its transmit, a deadlock situation may occur. To avoid this problem, when the channel is assumed to be assigned to a winner, each station activates a timer called the *idle-channel timer* (ICTimer). The duration of ICTimer must be greater than the duration of RCTimer. ICTimer is reset if incoming channel is sensed busy. However, if the incoming channel remains idle after ICTimer expires, it is assumed that the winner forfeits its transmission right. Each ready station then repeats the request contention operation to compete for the channel access right.

Finally, before a newly started station can join the network, it must wait and monitor the channel for a time period longer than the duration of ICTimer plus T_s. This will ensure that the station does not disrupt any ongoing events.

In Fig. 2, we construct a finite state machine to describe the detail operation of a RCMA transceiver.

3 Performance Analysis

3.1 The Model

Let a network consist of m stations. Each station is saturated so that it always has data to transmit. In other words, the event E1 shown in Fig. 2 occurs as soon as the station enters "idle" state.

We assume that the distance between any two stations is the same. We consider a realistic data frame size distribution. We assume that 35% of the data frames carry 46 bytes of useful information and 65% of the rest carry 1500 bytes of useful information, corresponding to the minimum and the maximum sizes of IEEE 802.3 frames [3].

We consider a cycle on the channel of RCMA shown in Fig. 3. Each cycle consists of the following: (i) an *I*-period; (ii) an *R*-period; (iii) a *C*-period; and (iv) a

D-period; representing an idle, request transmission, request collection and data frame transmission periods respectively.

The *I*-period starts as soon as the previous *D*-period ends. According to Fig. 2, in the *I*-period, all ready stations, including the station just completed a data frame transmission, enter the "request waiting" state. At this state, each station may transmit its request if its RWTimer expires. As soon as the first transmission of the request frame appears on the channel, the *I*-period ends and the *R*-period begins.

During the *R*-period, each station is not aware of any request frames transmitted by other stations, hence whenever its RWTimer expires, that station transmits its request frame. When the first bit of the first request frame reaches all stations, the *R*-period ends, and the *C*-period begins.

When the *C*-period begins, no further request frame can be transmitted. During this period the stations collect the requests to determine the winner. The winner initiates its data frame transmission when its RCTimer expires. The *C*-period ends when the winner starts its data frame transmission.

Due to the non-contention channel assignment operation, During the *D*-period, several data frame transmissions may occur. The *D*-period ends when there is no NEXT frame transmission after a data frame transmission.

3.2 Saturation Throughput Analysis

Let B be the data rate of the network. Given m saturated stations, let the random variables I, R, C, D be the duration of the *I*-period, the *R*-period, the *C*-period and the *D*-period respectively. Let the random variable U be the duration of the actual data transmission, excluding all IEEE 802.3 frame overheads during a cycle, and H be the duration of the overhead transmission such that $D=H+U$. Then the RCMA saturation throughput for m saturated stations, S_{RCMA}, can be expressed by

$$S_{RCMA} = \frac{E[U]}{E[I+R+C+H+U]}. \tag{1}$$

As described earlier, RWTimer=$w \cdot T_s$, where w is a uniformly distributed random integer between zero and k-1, and T_s is the minislot time duration. The *I*-period ends when at least one request frame transmission appears. Hence the probability that the *I*-period lasts for x minislots is the probability that any of the m stations choose to transmit their request frames given that no request frame transmission appears in previous minislots. The probability density function (pdf) of I is thus

$$P\{I = x \cdot T_s\} = \begin{cases} q_x & ,x=0 \\ q_x\left(1-\sum_{i=0}^{x-1} P\{I = i \cdot T_s\}\right) & ,x=1,2,\ldots,k-1 \\ 0 & ,x=k \end{cases} \tag{2}$$

where $q_x = 1-\left(1-\frac{1}{k-x}\right)^m$ is the probability that any of the m stations choose to transmit its request frame after x idle minislots.

For the *R*-period, since the distance between any two stations is fixed, the duration for a signal to propagate from any station to all stations is constant. Thus

$$R = \tau. \tag{3}$$

When the *R*-period ends, no further request frame can be transmitted. Since the *R*-period is a constant, then the number of minislots, *r*, within the *R*-period is also a constant, and it can be obtain by

$$r = \lfloor \tau / T_s \rfloor \tag{4}$$

where $\lfloor x \rfloor$ is the floor of *x*, defined as the largest integer smaller than *x*.

The duration of the *C*-period depends on the position of the winner within the *r* minislots. The value of the random variable *C* is between τ+T_s and 2τ. Since in LANs, the signal propagation time, τ, is generally small compared to the data frame transmission time (for example in our case, the data frame transmission for a long frame is about six times larger than τ), hence this random variable has only little effect on the saturation throughput of RCMA. Therefore, we here consider the worst case where the winner always appears at the last position during the *R*-period, that is

$$C = 2\tau. \tag{5}$$

Given *m* saturated stations, *r* minislots and the *k* parameter, the pdf of the number of request frames successfully detected by all stations during the *R*-period, *N*, can be derived recursively to be

$$P\{N = x\} = \frac{N_b(x,r,k,m)}{N_a(k,m)}, x = 0,1,...,r \tag{6}$$

where

$$N_b(x,r,k,m) = \binom{m}{0} N_b(x,r,k-1,m) + \binom{m}{1} N_c(x-1,r-1,k-1,m-1)$$
$$+ \sum_{n=2}^{m} \binom{m}{n} N_c(x,r-1,k-1,m-n),$$
$$N_c(x,r,k,m) = \binom{m}{0} N_c(x,r-1,k-1,m) + \binom{m}{1} N_c(x-1,r-1,k-1,m-1)$$
$$+ \sum_{n=2}^{m} \binom{m}{n} N_c(x,r-1,k-1,m-n),$$
$$N_a(k,m) = k^m,$$

with $\binom{m}{n} = \frac{m!}{n!(m-n!)}$ and the following initial conditions,

$$N_b(x=-1,r,k,m) = 0;$$
$$N_b(x=0,r,k \neq 1,m=0) = 1; N_b(x=0,r,k=1,m=1) = 0;$$
$$N_b(x=0,r,k=1,m \neq 1) = 1; N_b(x=1,r,k \neq 1,m=0) = 0;$$
$$N_b(x=1,r,k=1,m=1) = 1; N_b(x=1,r,k=1,m \neq 1) = 0;$$
$$N_b(x \geq 2,r,k \neq 1,m=0) = 0; N_b(x \geq 2,r,k=1,m) = 0;$$

and

$$
\begin{aligned}
&N_c(x=-1,r,k,m)=0;\\
&N_c(x=0,r=0,k,m)=N_a(k,m); N_c(x=0,r,k\neq 1,m=0)=1;\\
&N_c(x=0,r,k=1,m=1)=0; N_c(x=0,r,k=1,m\neq 1)=1;\\
&N_c(x=1,r=0,k,m)=0; N_c(x=1,r,k\neq 1,m=0)=0;\\
&N_c(x=1,r,k=1,m=1)=1; N_c(x=1,r,k=1,m\neq 1)=0;\\
&N_c(x\geq 2,r=0,k,m)=0; N_c(x\geq 2,r,k\neq 1,m=0)=0;\\
&N_c(x\geq 2,r,k=1,m)=0.
\end{aligned}
$$

$N_c(x,r,k,m)$ is the total number of possible permutations, that x out of r minislots will carry successful requests, given m and k. $N_b(x,r,k,m)$ is similar to $N_c(x,r,k,m)$ but $N_b(x,r,k,m)$ is the number of possible permutations under the assumption that no idle slot appears in any of the previous minislots. $N_a(k,m)$ is the total number of possible permutations given m and k.

The duration of the D-period depends on the number of successful requests appear in the R-period given in (6). If there was no successful request, all ready stations will enter the “request pending” state in Fig. 2 due to the events E6b and E8. Not all stations discover the failure of channel assignment at the same time, but the difference between the time each station enters the “request pending” state is not significant. Therefore we assume all stations return to the “request pending” state at the same time after the C-period ends. In the case where there is no winner, if ICTimer lasts for 2τ, then it will take duration of 2τ before this cycle ends. With this assumption, the relationship between the number of successful requests, N, obtained in (6) and the duration of the D-period is

$$
D=\begin{cases}
2\tau & ,N=0\\
E[T_{FRAME}]+\tau+2\cdot T_{IFG} & ,N=1\\
\sum_{i=1}^{N-1}\left(E[T_{FRAME}]+T_{IFG}+T_{NEXT}(i)+\tau\right)+E[T_{FRAME}]+\tau+2\cdot T_{IFG} & ,2\geq N\geq r
\end{cases}
\tag{7}
$$

with $T_{NEXT}(i)=8\cdot(10+7i)/B$, and T_{FRAME}, T_{NEXT} and T_{IFG} are the transmission time of the IEEE 802.3 frame, the NEXT frame and the IFG duration respectively. Knowing the distribution of a data frame, the mean of D can be computed.

The time duration of useful information transmitted during a cycle, U, also depends on N in (6). It can be expressed as

$$
U=N\cdot E[T_u],N=0,1,\ldots,r \tag{8}
$$

where T_u is the transmission time of the useful information. Having obtained the pdf of I, R, C, D, and U, their mean values can be computed, as well as the saturation throughput of RCMA given in (1).

4 Performance Comparison of RCMA and IEEE 802.3z

4.1 Saturation Throughput of IEEE 802.3z

The saturation throughput of Ethernet has been performed in [4]. Some modifications are made here to include the carrier extension operation of Gigabit Ethernet. From [4], the saturation throughput of IEEE 802.3z protocol, S_{CSMA}, is

$$S_{CSMA} = \frac{E[U_{CSMA}]}{E[I_{CSMA} + C_{CSMA} + H_{CSMA} + U_{CSMA}]} . \tag{9}$$

where the random variables I_{CSMA}, C_{CSMA}, H_{CSMA}, U_{CSMA} are the idle, contention, overhead transmission, and the useful information transmission periods respectively in CSMA/CD. Let D_{CSMA} be the frame transmission period including the overhead, thus $D_{CSMA}=H_{CSMA}+U_{CSMA}$. By [4]

$$\begin{aligned} &E[I_{CSMA}] = 0 \\ &E[C_{CSMA}] = (L_m - 1) \cdot T_{SCSMA} . \end{aligned} \tag{10}$$

where L_m, given in [4], is the mean number of slots required to resolve a collision caused by m stations and to obtain a successful transmission, and T_{SCSMA} is the slot time.

Given T_{IFG}, $T_{CARRIER}$, T_{FRAME} and T_u to be the duration of the IFG, the duration of carrier extension, the IEEE 802.3 frame transmission time, and the useful information transmission time respectively, the mean values of D_{CSMA} and U_{CSMA} are

$$\begin{aligned} &E[D_{CSMA}] = E[T_{FRAME}] + E[T_{CARRIER}] + T_{IFG} + \tau \\ &E[U_{CSMA}] = E[T_u] . \end{aligned} \tag{11}$$

By substituting (10) and (11) into (9), the saturation throughput of IEEE 802.3z can be obtained. It is important to note that due to the capture effect in the Ethernet protocol that results in temporary unfairness, a minor modification of the Ethernet protocol has been applied in [4] to eliminate the transient effect influencing the steady state results of the saturation throughput. The results here represent the worst case of the actual Gigabit Ethernet saturation throughput.

4.2 Performance Comparison

In this subsection, we compare the saturation throughput of RCMA and the IEEE 802.3z MAC protocol at 1Gb/s. The parameters used for numerical computations as well as computer simulations are based on [2, 3]. They are listed in Table 1.

We assume that the data frames consist of a mix of long and short frames with 35% of the frames being short, and 65% being long. Note that in IEEE 802.3z, if the data frame transmission duration (excluding preamble bits and SFD) is less than a slot time, the transmission will be extended with carriers until the duration of a slot time is reached. E[$T_{CARRIER}$] represents the average time wasted due to carrier extension for each data frame transmission with the assumed data frame distribution.

The saturation throughput of RCMA and the IEEE 802.3z MAC protocol are compared in Fig. 4. The analytical results (shown in lines) are also verified by the simulation results (shown in symbols). The analytical results for RCMA are slightly below the simulation results because we consider the worst case in the analysis.

Table 1. The parameters of RCMA and the IEEE 802.3z MAC protocol

Parameter	Value
Data rate, B	1Gb/s
Station numbers, m	1,2,…,50
Propagation delay, τ	2μsec
The IEEE 802.3 frame overhead including preamble and SFD	0.208μsec (26 bytes)
The useful transmission duration for a short IEEE 802.3 frame	0.386μsec (46 bytes)
The useful transmission duration for a long IEEE 802.3 frame	12μsec (1.5kbytes)
The IFG time duration, T_{IFG}	0.049μsec
The slot time in IEEE 802.3z, T_{SCSMA}	4.096μsec
The minislot time duration in RCMA, T_s	0.128μsec (16 bytes)
The minislot numbers in RCMA, r	15
The parameter k for RCMA	20

Comparing the throughput of the two protocols, the saturation throughput of IEEE 802.3z drops quickly when the number of saturated stations increased from one to five, and its throughput continues to drop as the number of saturated stations increases. The throughput even drops below 10% when there are over 32 saturated stations sharing the 1Gb/s bandwidth. In other words, each saturated station only receives at around 3.125Mb/s bandwidth on average under this condition.

On the other hand, the performance of RCMA is stable, it offers over 65% efficiency for up to 50 stations, except when the number of saturated stations is below three. This is because when the number of saturated stations is low, the channel assignment overhead for each data fame transmission is slightly higher due to the need for requests prior to data frame transmissions. However, as the number of saturated stations increases, the non-contention channel assignment operation of RCMA becomes effective, more data frame transmissions can be assigned during a request contention period, thus the channel assignment overhead for each data frame transmission becomes relatively small. In the case of 32 saturated stations, RCMA achieves around 70% throughput, which is equivalent to 21.875Mb/s bandwidth for each station on average, seven times higher than that in the IEEE 802.3z protocol.

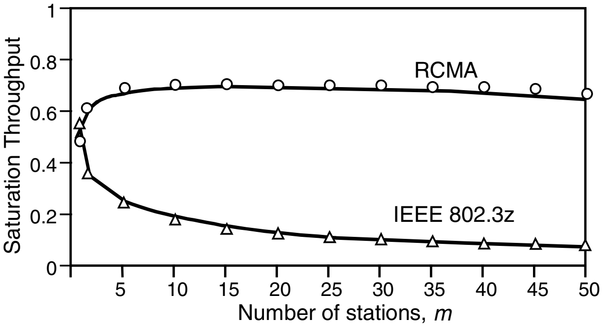


Fig. 4. The saturation throughput of RCMA and the IEEE 802.3z MAC protocol

Table 2. The duration of mean transmission time

Variable	Value
The mean frame transmission time, E[T_{FRAME}]	8.1368μsec
The mean useful information transmission time, E[T_u]	7.9288μsec
The mean duration of carrier extension in the IEEE 802.3z MAC protocol, E[$T_{CARRIER}$]	1.2544μsec

5 Conclusion

We have proposed a new MAC protocol for Gigabit LANs called RCMA. In RCMA, a sender uses the return signals repeated by a passive optical repeater as an acknowledgement to determine if its earlier transmission is successful. To make the protocol more efficient, the sender is required to contend for a channel access right for its data frame transmission using a very short request frame. This leads to a small bandwidth loss due to collisions of the request frames which is significantly small compared with the loss of bandwidth due to data frame collisions in IEEE 802.3z.

To further improve the performance of RCMA, the non-contention channel assignment operation was introduced. Under the non-contention channel assignment operation, the channel assignment information is explicitly passed by a sender to others to minimized the overhead of the channel assignment task. Moreover, RCMA can easily support service differentiation.

Finally, comparing the performance of RCMA and the current IEEE 802.3z MAC protocol, RCMA offers relatively stable and efficient performance under traffic saturation conditions while the performance of the IEEE 802.3z MAC protocol drops below 20% in the case of merely 10 saturated stations.

Since the performance of RCMA remains stable and efficient even if the number of saturated stations is as many as 50, we believe that the use of our proposed RCMA protocol in Gigabit LANs for the network access from a group of shared end users is not only cost efficient, but also far more reliable than the currently available standards and solutions.

References

1. Bertsekas, D., Gallager, R.: Data Networks, 2nd edition. Prentice Hall (1992).
2. Kadambi, J., Crayford, I., Kalkunte, M.: Gigabit Ethernet: Migrating to High-Bandwidth LANs. Prentice Hall (1998).
3. IEEE 802.3/ISO 8802-3: Information Processing Systems - Local Area Networks - Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications, 2nd edition, New York (1990).
4. Foh, C.H., Zukerman, M.: Performance Comparison of CSMA/RI and CSMA/CD with BEB. Proceeding of IEEE ICC 01, Helsinki (2001).

The Bluetooth Technology: State of the Art and Networking Aspects

Dajana Cassioli, Andrea Detti, Pierpaolo Loreti, Franco Mazzenga, and
Francesco Vatalaro

University of Rome Tor Vergata,
Dipartimento di Ingegneria Elettronica
and
RadioLabs
Consorzio Università Industria - Laboratori di Radiocomunicazioni
{surname}@ing.uniroma2.it

Abstract. Bluetooth is considered as a low-cost short-range wireless technology to provide communication functionalities, ranging from wire replacement to simple personal area network. In Bluetooth local networking applications a critical issue still under study is the evaluation of the network capacity when multiple piconets are simultaneously active in the same area, while providing mutual interference. In this paper we first provide a review of the main characteristics of the Bluetooth technology then we propose a semi-analytical approach to calculate the packet loss probability, and the aggregate network throughput. The analytical approach was validated by extensive comparison with simulation results showing a good agreement.

1 Introduction

The increasing importance of Internet web-based data applications and the pressing request for mobility pushed the research activities towards the definition of new global radio access networks.
Wireless personal area networks (PANs) represent the first access level to the global network. An equipment used for PAN communications should be low-cost, should provide communications among very different appliances, should interface with both wired and wireless external networks and should assure a relatively large traffic capacity. In a typical domestic or office environment the number of communicating appliances and/or terminals accessing to Internet can be quite large and their position cannot be easily predicted. Therefore a proper selection of a suitable wireless technique to connect them is mandatory. Furthermore to ensure full connectivity, this radio technology should be able to dynamically create and to manage ad hoc network(s) among the communicating terminals in the area.
The Bluetooth technology [1]-[6] is conceived as an effective low-cost solution to the many problems of PAN communications. In this paper we first provide a review of the main features of the Bluetooth technology and a brief discussion on

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 479–490, 2002.

the characteristics of the currently available market products. Then we obtain a closed form solution for the packet loss probability in a Bluetooth network composed of uncoordinated and interfering piconets. The formulation is valid under many different operating conditions and it is used to evaluate the overall network throughput. The paper is organized as follows. In Section 2 we provide a brief description of the main features of the Bluetooth technology and in Section 3 we summarize the main characteristics of the Bluetooth chip-modem currently available on the market. In Section 4 we obtain the packet loss probability and in Section 5 we define the Bluetooth network throughput. In Section 6 we validate the proposed approach comparing the theoretical results with simulation results. Finally, in Section 7 we draw our conclusions.

2 Bluetooth Technology

The Bluetooth was conceived to connect heterogeneous pieces of equipment, such as cellular phones, portable PCs, printers and, in the near future, domestic appliances, both among them and with external networks (mainly the Internet). These equipment and appliances contain a Bluetooth modem, commonly implemented as (possibly) low cost chip(s).
Three main fields of applications of Bluetooth technology have been identified:

- cable replacement among different appliances (point-to-point applications);
- access points to the Internet and to other external networks, both wired and wireless (point-to-multipoint applications);
- personal area networks.

The cable replacement application will render easier the interchange of data among different peripherals. Some examples are: the wireless connection between a cellular phone and the earphone, the connection of a mouse and a keyboard with the PC processing unit, etc. Since Bluetooth is intended as a de-facto standard, its adoption for cable replacement overcomes many problems related to cable standardization worldwide.

Bluetooth can be used to implement very low cost wireless access points, providing secure channels over which many transactions (information retrieval, automatic payment, etc.) can occur. Access points can also be used to provide access to other networks both wired (PSTN, xDSL etc.) and wireless (GSM, UMTS etc.).

Finally Bluetooth allows to realize low-cost and effective personal area networks. One typical scenario is a conference room where it would be possible to connect the video projector to the personal PC of the speaker and to record the (audio/video) compressed presentation in the attendant's PCs. PAN functionalities can find wide application in both domestic and office environments. Bluetooth devices enable the deployment of domotics services to render the environment intelligent and responsive to the different user needs. Another area for Bluetooth-based PANs is in the field of infomobility applications represent. In this case, one vehicle equipped with a PAN should be able to connect to any external network to gather any kind of information from the surrounding environment.

The basic unit of a Bluetooth network is the ”piconet” and a simple scheme is drawn in Fig.1. Within one piconet up to eight active terminals can be connected and communicate over a common channel.

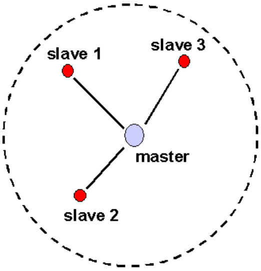


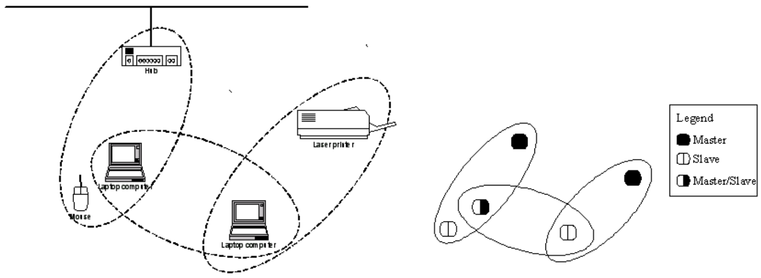
Fig. 1. Principle scheme of a piconet

The modulation format is GFSK and when a piconet is created one participant assumes the master role and controls the piconet operations. The remaining units in the piconet are indicated as slaves and can communicate only with the master in a time division duplex (TDD) fashion according to a polling sequence. [1]Multiple piconets in the area can share the ISM band through a frequency hopping spread spectrum (FH-SS) scheme. A maximum of 79 hopping frequencies are considered. Each piconet is given a frequency hopping pattern which identifies the common channel used by the master and the slaves in the same piconet to communicate. The pattern is dictated by the master and slaves are synchronized with this hopping sequence. A piconet can contain an undefinite number of ”parked ” (i.e. not active) terminals that constantly maintain the synchronization with the master. A single Bluetooth unit can participate in more piconets, but it can serve as a master only in one piconet. A set of intercommunicating piconets is commonly referred to as a ”scatternet”. An example of a scatternet is illustrated in Fig.2(a), where the roles of the different terminals in the three piconets have been depicted in Fig.2(b).

3 Bluetooth Technology: Market Status

With the release of the set of Bluetooth specifications 1.0b the manufacturers started to produce and to market the first Bluetooth development kits but it was observed that Bluetooth systems produced by different manufacturers were not

[1] In the current standard no routing functionalities are provided by the master to provide connections between two slaves in the piconet. This is a subject of current research and the BNEP [4] profile seems to provide an effective solution to this problem.



(a) (b)

Fig. 2. Principle scheme of a scatternet - (a): architecture, (b): network topology

perfectly compatible and there were problems concerning the proposed inquiry and paging procedures. Many problems were solved with the set of specifications 1.1, [1] and to achieve a Bluetooth specification conformity the products of the different manufacturers need to comply with a set of tests defined by Bluetooth special interest group.

The typical Bluetooth development kit includes a Bluetooth chip-modem mounted on a board. The Bluetooth chip-modem can be connected to a host sending control commands and data through a number of standard interfaces such as RS232/UART, USB and JTAG.

The architecture of a generic Bluetooth chip-modem can be partitioned into two parts: an RF part and a baseband (BB) stage as shown in Fig.3. The BB performs co-decoding and link control operations.

The control of the Bluetooth modem by the host is achieved through a set of standardized packet commands defining the "so-called" host controller interface (HCI). A Bluetooth chip-modem BB stage contains all the hardware and software (i.e. firmware) to receive, to decode and to process the HCI commands.

To facilitate the communication of a Bluetooth chip-modem with software applications, a number of high level communication functions have been standardized and grouped into different protocol layers: L2CAP, RFCOMM, SDP etc. The lowest layers functions use the services offered by the HCI commands.

Manufacturers propose different solutions for the Bluetooth chip-modem based on one or on two chips implementations. The CSR (www.csr.com) BlueCore 01 is a single Bluetooth chip-modem containing both the RF and the BB stages in Fig.3. The firmware in the BB part can be updated and is able to execute the HCI commands only. Manufacturers such as Philips (www.philips.com), OKI (www.oki.com), GCT (www.gctsemi.com), SiliconWave (www.siliconwave.com), propose two chips solutions where the RF and the baseband stages are physically separated.

In both cases (single chip implementation or two chip implementation) the pins of the Bluetooth chip(s) are directly connected to standard interfaces such as RS232 or USB or JTAG. This greatly simplifies the creation of boards containing the Bluetooth chips.

A single chip solution allows to reduce system dimensions and power consumption. However, due to the integration of the RF and the BB stages on the same

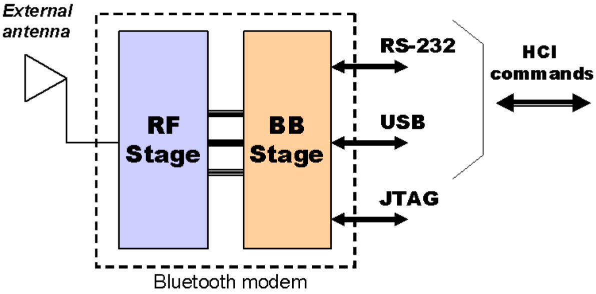


Fig. 3. architecture of a typical Bluetooth chip-modem

wafer the realization technology is much more expensive, thus increasing the cost of the final product. The adoption of separated processors allows to implement a more complex BB stage able to run also the desired application(s) as well as the upper layer protocols such as L2CAP, RFCOMM, SDP etc. This solution can be useful when a stand-alone Bluetooth embedded system is desired and the interfacing of the Bluetooth chip-modem with the device is completely controlled by the embedded system itself. However, as indicated by the CSR, the problems related to the scarce computation power in the baseband processor in the Bluecore 01 will be overcome in the next release of the chip indicated as Bluecore 02 so that it will be possible to implement a complete Bluetooth based system using a single chip.

4 Packet Loss Probability in Bluetooth Networks

Using different FH code patterns, several piconets can coexist in the same area indicated as served area, realizing a Bluetooth network The terminals aggregate randomly to form a large number of uncoordinated piconets with a different number of slaves. In a Bluetooth network packet collisions may occur when different piconets are transmitting simultaneously on the same frequency. This leads to an interference in the receiver (possibly) causing packet loss. Therefore, the packet loss probability (PLP) is an important performance index. PLP calculation needs to account for the dependency of the packet interference on the spatial distribution of terminals and on the environment characteristics. Results on this topic have been already presented in the literature [7]-[9]. An analytical approach for the PLP calculation was presented in [8], but results were restricted to a three overlapping piconets and a simple propagation model. In [9] a PLP upper bound is given without considering the "mitigation" effects due to propagation losses. In this section we provide a closed form expression for the PLP able to account for the geometry of the environment, its propagation characteristics and for the position of the reference receiver (RR).

We consider a Bluetooth network with $M + 1$ piconets. The PLP is commonly defined as the probability that the signal to interference plus noise ratio at the output of the RR falls below a threshold, ρ_0, which accounts for the fast fading characteristics of the environment:

$$P_{LP}(M) = Prob\left\{\frac{C}{\sum_{m=1}^{M}\chi_m Y_m + N} \leq \rho_0,\right\} \tag{1}$$

where C is the received power at the RR, N is the noise power. The $\sum_{m=1}^{M}\chi_m Y_m$ is the total interference power where Y_m is the interference power from the m-th piconet and χ_m is a binary random variable assuming with probability $q_m = 1 - p_m$. The p_m is the probability that the m-th piconet in the area transmits on the same frequency slot of the RR. Consider only one-slot packet transmission and assume that each slot always contains a packet. For the case in which packet duration is equal to the time slot, we have:

$$p_m = p = \begin{cases} 1/N_f, & \text{syncronized piconets} \\ 1-(1-1/N_f)^2 & \text{assyncronized piconets} \end{cases}, \tag{2}$$

where N_f is the number of hopping frequencies ($N_f = 79$). Other expressions of p in (2) for different values of the packet duration compared to the time slot are reported in [9].

The C and Y_m depends on the propagation losses due to the transmitter-receiver distance d and to the obstacles geometry. Therefore, indicating with W_T the terminal transmitted power assumed to be constant (i.e. no power control) we have: $C = W_T \times L_c$, and $Y_m = W_T \times L_m$. [2] The power losses L_c, L_m depend on the position of the RR and of the m-th interferer. When a stochastic propagation model is considered, L_m can be factored into two components, i.e. $L_m = A_m \times R_m$, where A_m is a deterministic component usually referred to as path loss and depends on the transmitter-receiver distance d; R_m is a random component accounting for shadowing. Since the power transmitted by each interfering user goes through the same propagation environment, the statistics of the interfering power measured at the RR are independent on m. Therefore, in the following we omit the dependency on m and we consider $Y = W_T \times L$. We restrict our analysis to a 2-D environment. Both C and Y are considered as random variables with probability density functions (p.d.f.) $f_C(x)$ and $f_Y(x)$ that in general depend on the position of the RR in the area. They can be evaluated as indicated in [10] using numerical approximations provided the statistics of $A = A_m$, $R = R_m$ and spatial distribution of the interfering terminals in the served area, are given.

From simple algebraic manipulations, equation (1) can be rewritten as:

$$P_{LP}(M) = Prob\{Z_M \leq 0\}, \tag{3}$$

with $Z_M = C - \rho_0 N - \rho_0 \sum_{m=1}^{M}\chi_m Y_m = Z_{M-1} - \varepsilon_M$ where $\varepsilon_M = \rho_0 \chi_M Y_M$ and $Z_0 = C - \rho_0 N$. For simplicity but without loss of generality in the following derivation we omit the noise power N. From (3) the p.d.f. of Z_M is:

$$f_{Z_M}(x) = f_C(x) \otimes f_{\varepsilon_1}(-x) \otimes \cdots \otimes f_{\varepsilon_M}(-x), \tag{4}$$

[2] No power control assumption is not valid for Bluetooth class 1 devices where power control is mandatory [1]

with $f_{Z_0}(x) = f_C(x)$. After some calculations (3) can be rewritten in a more compact form as:

$$P_{LP}(M) = \sum_{m=1}^{M} \binom{M}{m} q^{M-m} p^m \beta_m, \tag{5}$$

where $\beta_m = \int_{-\infty}^{0} g_m(x) \otimes f_{Z_0}(x)dx$ and $g_m(x) = \rho_0^{-m} f_{Y_1}(-x/\rho_0) \otimes \cdots \otimes f_{Y_M}(-x/\rho_0)$ for $m = 1, 2, \ldots, M$ and $g_0(x) = \delta(x)$. The coefficient $\binom{M}{m} q^{M-m} p^m$is the probability that m among the M interfering piconets are transmitting on the same frequency of the RR. The coefficients β_m in (5) account for the PLP reduction due to path loss and shadowing. In fact, it is straightforward to observe that β_m is always less than one for each m and increases with m approaching to one as m tends to infinity. In addition the coefficients β_m depend on the position of the RR and on the dimensions of the network area compared to the RR coverage area.

In the simple case of $\beta_m = 1$ for each m we obtain the upper bound reported in [9] i.e.:

$$P_{LP}(M) = \sum_{m=1}^{M} \binom{M}{m} q^{M-m} p^m \beta_m \leq 1 - q^M. \tag{6}$$

Equation (5) can be conveniently rewritten as:

$$P_{LP}(M) = 1 - q^M - \sum_{m=1}^{M} \binom{M}{m} q^{M-m} p^m (1 - \beta_m). \tag{7}$$

Equation (7) allows to obtain successive approximations by neglecting the terms corresponding to β_m close to unity. This can be useful for large M when it can be difficult to obtain a good numerical approximation of β_m.

5 Bluetooth Network Aggregate Throughput

In the definition of the aggregate network throughput we need to account for the variations of the PLP with the position of the RR. We assume that piconets in the area have a number of units L and that fixed length packets are used for transmission (for example DM1); the slaves are always transmitting towards the master in their time slots and the master has always something to transmit for all slaves; a round-robin baseband scheduling is considered and no signalling information is exchanged over the radio interface. On the basis of the previous assumptions, the master uses the 50% of the radio resource whereas the other 50% is fair shared among the $L - 1$ slaves. Hence, indicating with C the overall radio capacity in the piconet measured in packet per seconds, in an error free environment the master gets $C/2$, and each slave gets $C/(2(L - 1))$.

Now, we focus our attention on the throughput of a master/slave bidirectional connection in the presence of PLP and assuming M piconets in the area. We define the throughput of a single connection of the piconet as the mean number of packets successfully received in the unit time. In this case considering the master to slave direction, it is easy to prove the overall throughput (T_{MS}) in the piconet:

$$T_{MS}(M, x_1, y_1, \dots, x_{L-1}, y_{L-1}, L) = \frac{C}{2(L-1)} \sum_{l=1}^{L-1} \left(1 - P_{LP}(x_i, y_i, M)\right), \quad (8)$$

where (x_i, y_i) indicates the coordinates of the slaves and the dependence of the PLP on (x_i, y_i) has been evidenced.

Due to the Bluetooth mechanisms, a slave is allowed to transmit to the master only when it receives a packet from the master. So, when a master packet is lost, the return slot is wasted. Taking into account for this effect, the overall throughput of the piconet in the slave master direction can be written as:

$$T_{SM}(M, x_1, y_1, \dots, x_{L-1}, y_{L-1}, X_M, Y_M, L) =$$

$$= \frac{C}{2(L-1)} \sum_{l=1}^{L-1} \left(1 - P_{LP}(x_i, y_i, M)\right)\left(1 - P_{LP}(X_m, Y_m, M)\right). \quad (9)$$

where (X_m, Y_m) indicate the master coordinates. Hence, the total throughput of the piconet is:

$$T_P(M, x_1, y_1, \dots, x_{L-1}, y_{L-1}, X_M, Y_M, L) = T_{MS} + T_{SM}, \quad (10)$$

Fixing the position (Xm, Ym) of the master in the area and assuming that the slaves coordinates are randomly generated independently we can define the average throughput of the piconet as:

$$\overline{T}_P(M, X_M, Y_M) = \int T_P \prod_{l=1}^{L-1} g(x_l, y_l | X_m, Y_m) dx_l dy_l, \quad (11)$$

where the integral is extended to the master coverage area and $g(x_i, y_i|X_m, Y_m) = g(x_1, y_1|X_m, Y_m)$ with $i = 2, \dots, L-1$ is the probability density function of the slave coordinates given the master position. Due to the statistical independence of the slave coordinates using (10) it can be shown after some algebraic manipulations that (11) is independent on the number of slaves in the piconet. Considering M piconets in the served area we can define the average aggregate network throughput as:

$$T_N(M) = \int \overline{T}_P(M, X_M, Y_M) f(X_m, Y_m) dX_m dY_m, \quad (12)$$

where the integral in (12) is extended to the served area and $f(X_m, Y_m)$ is the p.d.f. of the master coordinates. Assuming a uniform distribution we have $f(X_m, Y_m) = M/S_a$ where S_a is the surface of the served area. The calculation of (12) is dependent on the environment characteristics of the served area not allowing to outiline general conclusions. Therefore, we resort to simple upper and lower bounds to give an idea on the throughput variability. Indicating with $P_{min}(M)$ and $P_{max}(M)$ the minimum and the maximum PLP in the served area, using these values in (10), we obtain:

$$T_N(M) \geq (MC(1 - P_{max}(M)) \left(1 - \frac{P_{max}(M)}{2}\right) \tag{13}$$

and

$$T_N(M) \leq (MC(1 - P_{min}(M)) \left(1 - \frac{P_{min}(M)}{2}\right). \tag{14}$$

6 Numerical Results

6.1 Validation of the Proposed Approach

To assess the effectiveness of equation (5) we performed a Monte Carlo simulation considering a typical operating scenario. In each trial we generate M masters uniformly located in a rectangular served area. Each master forms a piconet with N_s active slaves, where N_s is a random number, uniformly distributed between 1 to 7. Following the recommendations in [5], the transmitted power W_T is set to 0 dBm. We assumed the following dual slope model for path loss, [1]:

$$A(d) = \begin{cases} 40 + 20\log_{10}(d), & d \leq 8.5m \\ 25.3 + 36\log_{10}(d), & d > 8.5m \end{cases} \tag{15}$$

To simplify the calculation procedure we neglected the presence of noise and shadowing. Assuming a receiver sensitivity of -70dBm using (8), the RR coverage area is circular with radius 10 m [2]. The N_s slaves participating in a piconet are randomly located according to a uniform distribution, in a circular area of 20 meters diameter, centered in the position of the piconet master. In generating the slaves' positions we ensured that they were confined within the served area. The transmitter to the RR is randomly located in a circular coverage area centered in the RR with 10 m radius. In this case when the piconets interference is null we assume that no packet loss occurs. Each piconet transmission begins in a randomly selected time slot. In every piconet the master begins the transmission by sending an ACL packet to one of the N_s slaves belonging to its piconet. For each master we randomly generated its own channel hopping sequence assuming a uniform distribution over the 79 frequency carriers. The length of the frequency hopping sequence for each master was taken equal to the duration of the simulation trial. We averaged the performance metrics over a large number of simulation trials for each scenario. In each trail we changed the users' positions and the piconet loads (the number of slaves in each piconet). In each time slot we compute the signal-to-interference ratio (C/I) of the RR.
In Fig.4 we compare the $P_{LP}(M)$ obtained from (5) with the results of a Monte Carlo simulation obtained for different dimensions of the served area supposed

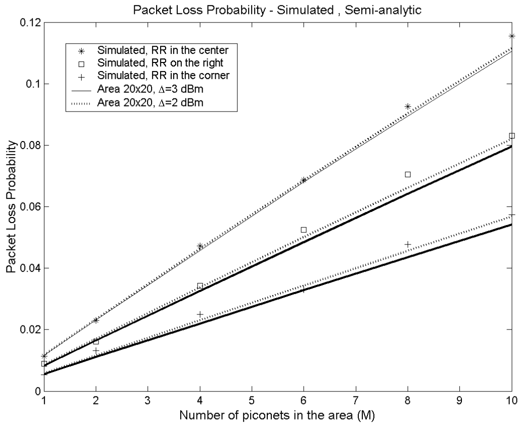


Fig. 4. Packet Loss Probability vs the number of piconets in the area - Area dimensions: 20 × 20 m - synchronous piconets

to be rectangular and for different positions of the RR. Discrete numerical approximation of the p.d.fs. $f_C(x)$ and $f_Y(x)$ were used to obtain the coefficients β_m in (5) (see [10]). The discrete step Δ of the random variable x is indicated in Fig.4. As expected the agreement between the simulated and theoretical results improves reducing Δ at the expense of an increased computation time. From the results in Fig.4 it can be observed that PLP significantly changes with the position of the RR in the served area.

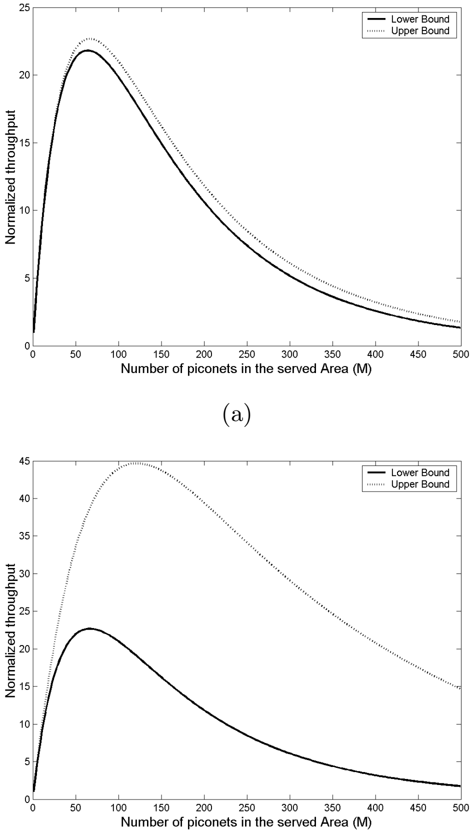
6.2 Network Throughput Calculation

We consider a rectangular served area, and the propagation model in (15). In this case it is straightforward to observe that the minimum PLP is obtained when the RR is placed in the corner and the maximum PLP is obtained when the RR is placed in the center. We introduce the normalized aggregate throughput $S(M)$ defined as $S(M) = T_N(M)/C$. In Fig.5 we plot the upper and lower bounds of the normalized scatternet throughput as a function of the number of piconets and for different dimensions of the served area. To evaluate the PLP we used equation (5) and the corresponding values of β_m are reported in Table 1. To obtain the results in Fig.5 and in Table 1 a uniform distribution of the interferers in the area was assumed.

From Fig.5 we observe that increasing in the number of piconets, the average normalized Bluetooth network throughput increases as well until a critical value of piconets is reached. Over this value the interference due to the higher number of collisions reduces the overall throughput. When the area is small compared to the coverage area of the RR, it is possible to adopt the PLP upper bound in (6) to evaluate the network throughput.

Table 1. Values of β_m used in (5) - RR in the center and in the corner of the served area

m/Area	**10m x 10m**		**20m x 20m**		**20m x 10m**	
	Center	Corner	Center	Corner	Center	Corner
β_1	0.9449	0.9115	0.9112	0.4020	0.9238	0.6208
β_2	0.9848	0.9770	0.9771	0.7683	0.9802	0.8979
β_3	0.9895	0.9855	0.9856	0.8665	0.9877	0.9479
β_4	0.9931	0.9903	0.9904	0.9209	0.9917	0.9695
β_5	0.9947	0.9922	0.9923	0.9449	0.9934	0.9823
β_6	0.9975	0.9952	0.9948	0.9566	0.9965	0.9902



(a)

(b)

Fig. 5. Normalized Bluetooth network throughput vs the number of piconets - upper and lower bounds area dimensions: (a): 10mx10m, (b): 20mx20m

This is no longer true for large areas where the difference between the upper and lower bounds of the throughput can be relevant (see Fig.5b) since the beneficial effects due to environments cannot be neglected.

7 Conclusions

The Bluetooth transmission technology is a viable and low cost solution to fulfill many of the requirements of short range wireless communications. In this paper we reviewed the technological aspects of the Bluetooth products available on the market and we discussed the advantages and drawbacks of one or two chips implementation. We analyzed the performance of a Bluetooth network consisting of a set of uncoordinated Bluetooth terminals in the same area in terms of PLP and aggregate network throughput. We obtained a closed form expression for the PLP accounting for the beneficial effects due to propagation losses. The effectiveness of the proposed calculation was assessed through a comparison with simulation results. As expected the PLP is strongly dependent on the position of the RR and on the size of the served area compared to the coverage area. Large variations of the upper and lower bounds of the network aggregate throughput with the PLP have been evidenced.

References

1. Specification of the Bluetooth System: Core, Version 1.1. February 22, 2001.
2. J. C. Haartsen, S. Mattison, "Bluetooth-A New Low-Power Radio Interface Providing Short-Range Connectivity", Proc. IEEE Vol. 88, No. 10, October 2000, pp.1651-1661
3. Jaap C. Haartsen, "The Bluetooth Radio System", IEEE Pers. Comm., February 2000, pp.28-36
4. Official Bluetooth website: http://www.bluetooth.com
5. P. Johansson, M. Kazantzidis, R. Kapoor, M. Gerla, "Bluetooth: an enabler for personal area networking", IEEE Network, Vol. 15, Issue: 5 Sept.-Oct. 2001, pp. 28-37
6. Bhagwat, P. "Bluetooth: Technology for Short-range Wireless Applications", IEEE Internet Computing , Volume: 5 Issue: 3 , May-June 2001 Page(s): 96 -103
7. Y. Lim, J. Kim, S. L. Min and J. Soo Ma, "Performance evaluation of the Bluetooth-based public internet access point," IEEE, 2001
8. A. Karnik and A. Kumar, "Performance analysis of the Bluetooth physical layer," Proceedings of the IEEE ICPWC, 2000.
9. A. El-Hoiydi, "Interference between Bluetooth Networks - upper bound on the packet error rate," IEEE Commun. Lett., vol. 5, No. 6, June 2001
10. F. Mazzenga, D. Cassioli, P. Loreti, F. Vatalaro, "Evaluation of Packet Loss Probability in Bluetooth Networks", to be presented in ICC 2002, April 28- May 2, 2002, New York City, USA

Time and Frequency Synchronization for Hiperlan/2

Anna Berno[1] and Nicola Laurenti[2]

[1] Dipartimento di Elettronica ed Informatica, Università di Padova
35131 Padova, Italy.
Now with Fracarro Radioindustrie, 31033 Castelfranco Veneto, Italy
[2] Dipartimento di Elettronica ed Informatica, Università di Padova
nil@dei.unipd.it
http://www.dei.unipd.it/~nil/index.html

Abstract. The Hiperlan/2 standard [1]-[3] for wireless LAN transmission in the 5 GHz frequency band makes use of OFDM modulation with a TDMA access scheme, in order to efficiently exploit time dispersive channels with frequency selective fading.
It is well known that the performance of OFDM schemes is very sensitive to synchronization: symbol timing and carrier frequency errors must be carefully estimated and corrected at the receiver.
We propose a scheme for time and frequency offset estimation, derived form those presented in [4]-[7], suited to all the transmission burst types of the standard. The scheme makes use of the periodic structure of each burst preamble and is robust with respect to distortions induced by dispersive channels.
We evaluate its performance both via statistical analysis and simulation in the presence of AWGN and dispersive channels, and also present an original technique for performance evaluation of the timing synchronization in dispersive environments, based on the cumulative distribution function of the useful signal power after demodulation.

1 Introduction

The Hiperlan/2 standard [1]-[3] aims at providing high bit rate wireless links to fixed or portable terminals within a local (mainly indoor) environment, with a channel bandwidth of 20 MHz in the 5 GHz band. It makes use of an OFDM modulation technique with cyclic prefix which makes transmissions very robust to dispersive channel affected by frequency selective fading, thus increasing its spectral efficiency, but is very sensitive to timing and frequency offsets between transmitter and receiver, which can cause intersymbol and intercarrier interference in the demodulated signal. It is therefore mandatory that time and frequency offsets be carefully estimated and corrected at the receiver.

We present a time and frequency synchronization scheme derived from [4]–[7], based on periodic preambles and adapted to suit Hiperlan/2 burst types. Since it is based on the signal periodicity rather than its actual expression, and

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 491–502, 2002.

periodicity is preserved through the channel, the algorithm effectively can work also in dispersive environment. Its performance is evaluated both in AWGN and dispersive channels by means of statistical analysis and simulation results. In particular we introduce an original method to evaluate the time synchronization performance in the case of a dispersive channel in terms of the power of the useful component in the demodulated signal.

The paper is organized as follows. In Section 2 we set up the system model with time and frequency offsets and a non ideal channel and in Section 3 we discuss the effects of time and frequency offsets on the performance of OFDM systems in an ideal or dispersive channel. In Section 4 we present the estimation algorithms and the techniques for analytical evaluation of their performance. Section 5 collects and discusses results obtained from simulations together with those derived analytically. Eventually we draw conclusions in Section 6.

2 System Model

OFDM parameters for Hiperlan/2 [3] are summarized in Table 1. The OFDM symbols are concatenated into the payload; a preamble is preponed to the payload and the two together form a physical layer (PHY) burst. Five different PHY burst types are provided [3], each corresponding to a different transmission mode: *broadcast, downlink, uplink with short preamble, uplink with long preamble, direct link* (optional). The preambles structure is illustrated in Fig. 1.

Table 1. OFDM parameters for Hiperlan/2

Parameter	Value
Sampling rate $F_0 = 1/T$	20 MHz
Carrier central frequency f_c	5.2 GHz
FFT size N	64
Useful symbol part duration T_U	$64T = 3.2\mu$s
Cyclic prefix duration T_{CP}	$16T = 0.8\mu$s (optional $8T = 0.4\mu$s)
Symbol interval $T_S = T_U + T_{CP}$	$80T = 4.0\mu$s (optional $72T = 3.6\mu$s)
Number of data sub-carriers N_{SD}	48
Number of pilot sub-carriers N_{SP}	4
Total number of sub-carriers $N_{ST} = N_{SD} + N_{SP}$	52
Sub-carrier spacing $F = 1/T_U$	0.3125 MHz
Nominal bandwidth $B = N_{ST}F$	16.25 MHz
Data symbol constellations	BPSK, QPSK, 16-QAM, 64-QAM

Relatively to transmission, the system can be modeled as in Fig. 2. Data and pilot symbols modulate the N_{ST} active subcarriers with indices in $\mathcal{M} = \{-N_{ST}/2, \ldots, -1, 1, \ldots, N_{ST}/2\}$, giving the modulated signal

$$s(t) = \sum_{m\in\mathcal{M}} \sum_{l=-\infty}^{+\infty} S_m(lT_S)p(t - lT_S)\, e^{j\,2\pi mF(t-lT_S)} \tag{1}$$

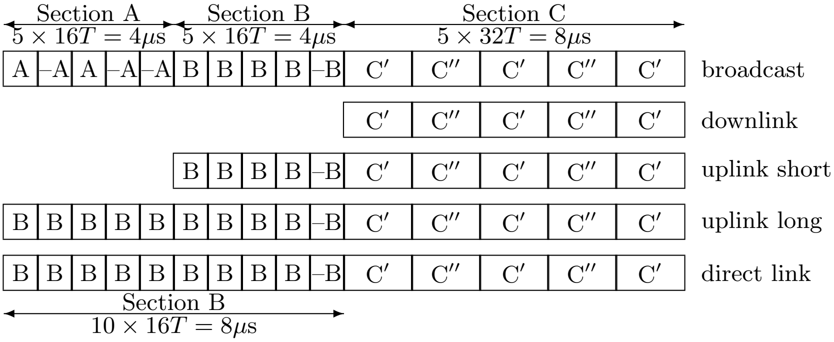


Fig. 1. Structure of Hiperlan/2 preambles for all burst types.

with $p(t)$ the rectangular window on the interval $[-T_{\mathrm{CP}}, T_{\mathrm{U}})$ and S_m the transmitted complex symbols.

The channel impulse response and the additive noise can be replaced by their baseband equivalents. The former can be modeled as a tapped delay line [9], with non-uniform time spacing between taps and an exponentially decaying power delay profile. A Doppler spread of 52 Hz is assumed for each tap, corresponding to a terminal speed of 3 m/s, so that the channel coherence time results $\tau_c \simeq 20$ ms, while each burst is always shorter than 2 ms. The channel can therefore be considered time-invariant for the duration of a burst and its impulse response is written as

$$h(\tau) = \sum_{k=0}^{N_h-1} a_s\, \delta(\tau - \tau_s), \tag{2}$$

where τ_s are the delays, and a_s the complex amplitudes. BRAN defined five channel models for Hiperlan/2 simulations: A and B, with a delay spread shorter than the duration of the cyclic prefix; C, D and E, with longer delay spread [8].

With $w(\cdot)$ the baseband equivalent noise, the received signal before sampling is $y(t) = [s * h(t)]e^{j\,2\pi \Delta f t} + w(t)$, with $\Delta f = f_0 - f_0'$ the carrier frequency offset between transmitter and receiver. If the starting instant of the FFT demodulating window is at t_0 we can then write the demodulated signal as

$$Y_m(kT_{\mathrm{S}}) = \sum_{n=0}^{N-1} y(t_0 + kT_{\mathrm{S}} + nT)e^{-j\,2\pi mn/N} \ . \tag{3}$$

The mismatch that can possibly turn up between the transmitter and receiver oscillators in the forms of *carrier frequency offset*, *phase noise in the RF oscillators*, *sampling or clock frequency offset*, *sampling or clock jitter*, or *OFDM symbol timing error*, is a common cause of impairment in an OFDM system [10].

In the course of our work only carrier frequency offset and symbol timing error will be considered with reference to a Hiperlan/2 system. As regards sampling frequency errors, it has to be noted that a quantification of their effect leads

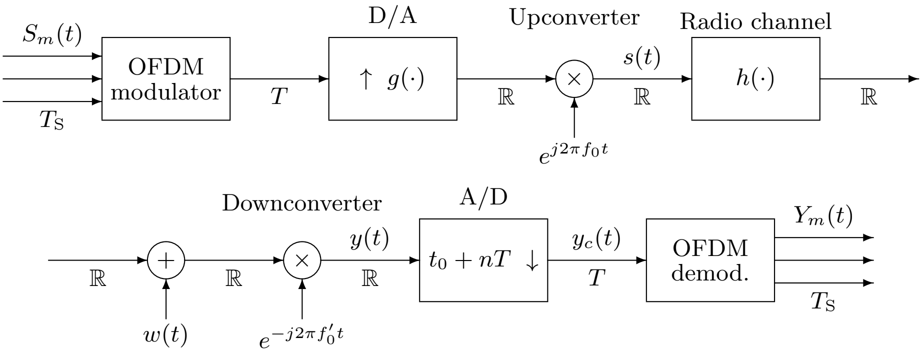


Fig. 2. Simplified model of an OFDM system with time and frequency offsets

to the conclusion that the maximum symbol phase rotation that can affect a Hiperlan/2 system approximates about 0.117 degrees if we consider an oscillator with a frequency instability of 10 ppm. As a consequence, the effects induced on the system by a sampling frequency offset have been neglected in our work.

3 Effects of Time and Frequency Offsets

3.1 Carrier Frequency Offset

A visible effect of a carrier frequency offset Δf [4] on the received symbols is a rotation of the received constellation of a phase equal to $2\pi\, t\, \Delta f,\ t \in Z(T_S)$. Due to Δf a shift of the received spectrum on the frequency axis and consequently a loss of mutual orthogonality between the subcarriers occurs. This results in InterChannel Interference (ICI). The variance of the ICI process, $\sigma^2_{\mathrm{ICI}_k}$, is the sum of the variances of the interference contributions:

$$\sigma^2_{\mathrm{ICI}_k} = \sum_{m \neq k} \sigma^2_{S_m} |H(f_m)|^2 \cdot \mathrm{sinc}^2(f_m - f_k + \Delta f), \tag{4}$$

with $H(f)$ the channel frequency response. The statistical properties of the ICI were evaluated in [10] showing that, contrary to what stated in [4] a Gaussian distribution can not be assumed in general, but it represents a fair approximation for dense constellations and small Δf.

3.2 Symbol Timing

A shift of the FFT observation window at the receiver with respect to the transmission window exceeding the guard interval causes samples from the previous or following OFDM symbol to fall within the current window (resulting in Inter-Symbol Interference, ISI) and samples of the useful part of the current symbol to be lost (resulting in loss of orthogonality among the subchannels and thus ICI). The optimum positioning of the observation window should take advantage of the margin left by the presence of the cyclic prefix. This margin is reduced, however, by a time-dispersive channel, because the first samples of the prefix and the last samples of the data might suffer from the interference due to delayed or early replicas of the adjacent OFDM symbols. On the other hand, ICI and ISI are unavoidable if the channel impulse response length exceeds the guard interval length.

In the presence of a time-dispersive channel $h(t)$, the input to the demodulating FFT window will in general contain some interference from adjacent OFDM symbols. Correspondingly to the channel frequency response at frequency mF, $H(mF)$, the demodulated symbol on the m-th subcarrier will be therefore affected by an amplitude and phase distortion, as well as by ICI and ISI. In this case, since the channel distortion and the consequent need for amplitude and phase equalization are unavoidable, finding the optimum timing means choosing the starting instant for the FFT demodulation window such that the power of the useful component of the signal is maximized. If we neglect noise, each output signal $Y_m(t)$ can be written as:

$$\begin{aligned} Y_m(kT_\mathrm{S}) = \varPhi_{m,m}(kT_\mathrm{S}, kT_\mathrm{S})\, S_m(kT_\mathrm{S}) + \sum_{r \neq m} \varPhi_{r,m}(kT_\mathrm{S}, kT_\mathrm{S})\, S_r(kT_\mathrm{S}) \\ + \sum_{l \neq k} \sum_{r} \varPhi_{r,m}(lT_\mathrm{S}, kT_\mathrm{S})\, S_r(lT_\mathrm{S}), \end{aligned} \quad (5)$$

with

$$\varPhi_{r,m}(lT_\mathrm{S}, kT_\mathrm{S}) = \frac{1}{N} \sum_{n=0}^{N-1} \mathrm{W}_N^{-nm} \int_{-T_\mathrm{CP}}^{T_\mathrm{U}} h(t_0 + nT + kT_\mathrm{S} - lT_\mathrm{S} - v)\, e^{j2\pi rFv} dv. \quad (6)$$

The second and third term of (5) represent the interference (respectively ICI and ISI) experienced by the demodulated OFDM symbol. On the other hand, under the simplifying assumption that $S_r(lT_\mathrm{S})$ are i.i.d. QAM symbols, with $\mathrm{E}[S_m] = 0$ and $\mathrm{E}[|S_m|^2] = M_S$, the term $M_\mathrm{u}(m,k) = M_S\, |\varPhi_{m,m}(kT_\mathrm{S}, kT_\mathrm{S})|^2$ is the power of the useful signal component with the FFT window starting at t_0. Consider the multipath channel impulse response 2, M_u can be evaluated as:

$$M_\mathrm{u} = \frac{M_S}{N^2} \Big| \sum_{s=0}^{N_h - 1} a_s \gamma(\tau_s - t_0)\, e^{-j2\pi mF\tau_s} \Big|^2, \quad (7)$$

with

$$\gamma(t) = \begin{cases} N + \lceil t/T \rceil & -T_\mathrm{U} < t \leq -T \\ N & -T < t \leq T_\mathrm{CP} \\ N - \lceil t/T \rceil + N_\mathrm{CP} & T_\mathrm{CP} < t < T_\mathrm{S} \\ 0 & \text{elsewhere} \end{cases} \quad (8)$$

The quantity M_u can be used to evaluate the symbol timing estimators performance in the presence of a time-dispersive channel. If M_u is normalized to $|H(mF)|^2$, we obtain $\overline{M}_\mathrm{u}$. Therefore, the closer $\overline{M}_\mathrm{u}$ approaches unity, the better timing estimation is performed.

4 Synchronization Algorithms

Synchronization methods for OFDM can follow either a pilot-aided or a blind approach. The presence of burst preambles in the transmitted Hiperlan/2 signal calls for the former approach, that is generally faster, while the latter relies on long range signal statistics.

In turn, pilot-aided estimation can be accomplished at the receiver either in the time domain (operating on the received signal prior to FFT demodulation), or in the frequency domain (i.e. operating on the demodulated signal). However, since the A and B preamble sections do not exhibit the cyclic prefix structure, the demodulated signal would suffer from loss of orthogonality in the presence of an even slightly dispersive channel, which makes frequency domain techniques less viable. Among time-domain methods, the repetitive structure of the preambles suggests that an algorithm such as the one proposed by Hanzo and Keller [4] can be modified to be adapted to Hiperlan/2 preambles, to achieve frequency and timing estimation without knowledge of the adopted reference sequence. Likewise, it is possible to think of the sections that compose all the preambles as if they were made by two identical symbols in the time domain, as required by Schmidl and Cox algorithm [6]-[7].

4.1 Symbol Timing Estimation

The Schmidl and Cox algorithm is modified considering that the received preamble signal $r(t)$ keeps its periodic structure, with period $T_p = LT$, inside the observation interval $(L + C)T$, that is $r_{m+L} = r_m$ for $m = m_0, \ldots, m_0 + C$. The timing metric for symbol synchronization $M(d)$, proposed in [6], is modified as

$$M(d) = \frac{\sum_{m=0}^{C_1-1} r^*_{d+m}\, r_{d+m+L} + \sum_{m=C_1}^{C_1+C_2-1} r^*_{d+m}\left(-r_{d+m+L}\right)}{\sum_{m=0}^{C-1} |r_{d+m+L}|^2}. \tag{9}$$

Expression (9) can be applied to all Hiperlan/2 burst preambles with different values for the parameters, correlates samples at a distance LT within a sliding window of length $(C + L)T$, thus overcoming the restriction imposed in [6] on the window length, which could include only one OFDM symbol. The optimum timing d_{opt} can be found by maximizing the metric (9): the parameter values for each burst type are given in Table 2.

On an ideal channel the performance of the timing estimator can be assessed in terms of its probability mass distribution, as d_{opt} is a discrete variable. With a dispersive channel, however, it is more appropriate to evaluate the power of the useful signal component (7) obtained with the estimated correct positioning of the FFT window.

4.2 Carrier Frequency Offset Estimation

Making use of the optimal timing information, frequency synchronization can be performed as if no timing error affected the system. Moreover, following the hint given in [4]-[5], an important effect can be achieved: the use of a training sequence with a *shorter* periodicity allows the frequency offset estimation range to be widened. Each OFDM symbol of Hiperlan/2 preambles is composed of $T_{\mathrm{U}}/T_{p'} = K$ short symbols, which can be considered identical except for a sign inversion, they have a minimum period of length $T_{p'} = T_{\mathrm{U}}/K$. The carrier frequency offset is estimated as:

$$\widehat{\Delta f} = \frac{1}{2\pi LT} \arg\Big(\sum_{m=0}^{L-1} \alpha\, r_{m+d_{\mathrm{opt}}}\, r^*_{m+d_{\mathrm{opt}}+L}\Big), \tag{10}$$

whit d_{opt} the optimum symbol timing and α and L depend on the preamble structure and are given in Table 2. The maximum frequency offset that can be detected becomes $\Delta f_{\max} = 1/(2LT) = NF/(2L)$. As an example, for the broadcast burst, the initial frequency offset required is $\Delta f_{\max} = 0.625$ which yields an upper bound to the oscillators stability of about 120 parts per million (ppm), a condition easily met by commercial devices.

Table 2. Parameters of the symbol timing and carrier frequency offset estimators for the different burst types

		Symbol timing				Frequency offset		
Burst type	Preamble section	L	C_1	C_2	C	α	L	$\Delta f_{\max}$
Broadcast	A	32	32	0	32	-1	16	$2F = 0.625$ MHz
Downlink	C	64	96	0	96	1	64	$F/2 = 0.156$ MHz
Uplink short	B	32	32	0	32	1	16	$2F = 0.625$ MHz
Uplink long	B	80	64	16	80	1	16	$2F = 0.625$ MHz
Direct link	B	80	64	16	80	1	16	$2F = 0.625$ MHz

The statistical evaluation of the frequency offset estimator can be made in terms of its conditional mean and statistical power, that is $\mathrm{E}[\widehat{\Delta f}|\Delta f]$ and $\mathrm{E}[\widehat{\Delta f}^2|\Delta f]$. Our analysis is brought on in the hypothesis of perfect symbol timing correction. The received signal $r(nT)$ can be written as $r_{n+d_{opt}} = s_n\, e^{j2\pi nT\Delta f} + w_n$, with s_n the transmitted OFDM signal and $w_n \in \mathcal{N}C(0, 2\sigma^2)$ complex gaussian noise. With the simplifying hypothesis of high signal to noise ratio we can write

$$\sum_{m=0}^{L-1} r^*_m\, r_{m+L} \simeq \sum_m |s_m|^2 + \sum_m (s^*_m\, \eta_{m+L} + s_m\, \eta^*_m) = S + W, \tag{11}$$

with $S = \sum_m |s_m|^2$ and $W \in \mathcal{N}C(0, 4\sigma^2 S)$. Approximating (10) with $\widehat{\Delta f} = \arg(S+W)/(2\pi LT)$ and following [11] the conditional PDF of $\widehat{\Delta f}$ for $\lambda \in \big[-\frac{1}{2LT}, \frac{1}{2LT}\big]$ is:

$$f_{\widehat{\Delta f}|\Delta f}(\lambda|\mu) = 2\pi LT\Big(c_0 + \sum_{n=1}^{+\infty} c_n\, \cos n2\pi LT(\lambda - \mu)\Big). \tag{12}$$

with $\rho = S^2/\mathrm{E}[|W|^2] = L\left\langle |s_m|^2\right\rangle/2\mathrm{E}[|w_m|^2]$, where $\left\langle |s_m|^2\right\rangle$ is the mean of the deterministic sequence $|s_m|^2$ in the time domain, and

$$c_0 = \frac{1}{2\pi},\ c_n = \frac{1}{2\sqrt{\pi}}\sqrt{\rho}\,e^{-\frac{\rho}{2}}\left[I_{\frac{n-1}{2}}\left(\frac{\rho}{2}\right) + I_{\frac{n+1}{2}}\left(\frac{\rho}{2}\right)\right]. \tag{13}$$

With a term by term integration we have

$$\mathrm{E}[\widehat{\Delta f}|\Delta f] = \frac{1}{LT}\sum_{n=1}^{+\infty}\frac{(-1)^{n+1}c_n}{n}\ \sin(n2\pi LT\Delta f) \tag{14}$$

$$\mathrm{E}[\widehat{\Delta f}^2|\Delta f] = \frac{1}{12(LT)^2} + \frac{1}{\pi(LT)^2}\sum_{n=1}^{+\infty}\frac{(-1)^n c_n}{n^2}\ \cos(n2\pi LT\Delta f) \tag{15}$$

$$\mathrm{Var}[\widehat{\Delta f}|\Delta f] = \mathrm{E}[\widehat{\Delta f}^2|\Delta f] - \{\mathrm{E}[\widehat{\Delta f}|\Delta f]\}^2. \tag{16}$$

5 Simulation Results

Simulations have been carried out for each of the five burst types supported by Hiperlan/2, with an AWGN channel as well as with the channel models in [8], at different SNR conditions. Here we discuss the main results.

5.1 Synchronization Performance with AWGN Channel

Symbol Timing Estimation. The results for the symbol timing estimators are illustrated in Fig. 3 in terms of histograms of the estimation error.

Broadcast Burst. The performance of the symbol timing estimator is very good even at low signal to noise ratio, since even with SNR = 5 dB the percentage of correct estimations reaches 80% of the total simulated transmissions.

Downlink Burst. The mass distribution presents an evident bias towards a delay of one or more samples in the timing estimation. In our opinion this is due to the fact that correlation between the samples falling in the two halves of the observation window in use for the calculation of the timing metric periodically happens to be greater when the window is filled with one or more samples of the OFDM symbol that follows the preamble, than when it is correctly placed on the preamble only.

Uplink Burst with Short Preamble. Symbol timing performs well also in the case of an uplink burst with short preamble. Even at SNR = 5 dB in fact the percentage of correct timing decisions reaches 70% of the total simulated transmissions and grows to more than 90% at SNR = 10 dB.

Uplink Burst with Long Preamble and Direct Link Burst. The performance of the algorithm is extremely good in this case, due to the fact that the observation window is longer than in the other cases, thus reducing the estimation variance. The percentage of correct estimations equals almost 100% even at SNR = 5 dB.

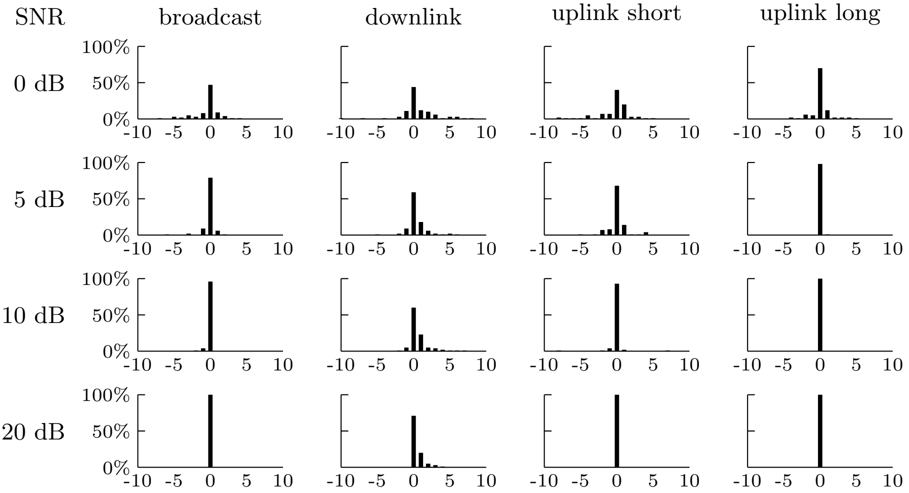


Fig. 3. Histograms of symbol timing estimation error for all burst preambles at different SNR values

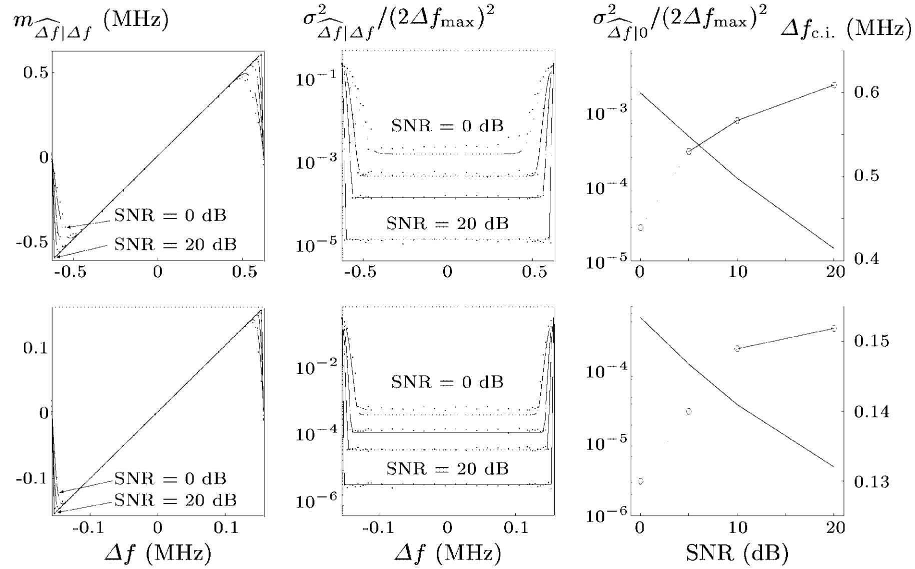


Fig. 4. Statistics of the carrier frequency offset estimator using the broadcast (top) and downlink (bottom) burst preambles. On the left and in the center: conditional mean and normalized conditional variance versus true offset in the range $[-\Delta f_{\max}, \Delta f_{\max}]$ for SNR = 0,5,10,20 dB; on the right: normalized variance in the coherence interval (marked with ×, log scale on the left axis) and width of the coherence interval (marked with ○, linear scale on the right axis) versus channel SNR.

Carrier Frequency Offset Estimation. The performance of the carrier frequency offset estimator has been evaluated through its mean and variance conditioned on the actual value of the carrier frequency offset Δf, in the range $[-\Delta f_{\max}, \Delta f_{\max}]$.

Typically, the estimator can be considered *practically unbiased* within a range $[-\Delta f_{\text{c.i.}}, \Delta f_{\text{c.i.}}]$, named *coherence interval* (c.i.) and within this range, the estimator variance is nearly constant. Outside the c.i. the estimation presents a bias towards the origin and its variance rapidly grows, so that the estimation is clearly not reliable. Therefore, $\Delta f_{\text{c.i.}}$ represents the maximum carrier frequency offset that can reliably be estimated. As the SNR increases, the estimator variance in the c.i. decreases in an inversely proportional fashion, while the width of the c.i. grows towards its asymptotic value $\Delta f_{\max}$.

In Fig. 4 we show the conditional statistics and coherence interval of the estimator for the broadcast and downlink bursts. The simulation results are depicted with a dot notation, while the solid line represents the theoretical curve given by (14) and (16). The results for the uplink burst with short and long preamble and the direct link bursts are similar to those for the broadcast burst.

Broadcast Burst. In this case the maximum detectable offset $\Delta f_{\text{c.i.}}$ is seen to extend up to 0.42 MHz = 80 ppm at SNR = 0 dB, and to 0.6 MHz = 115 ppm at SNR = 20 dB. The estimator variance, normalized to $(2\Delta f_{\max})^2$ goes from $2.47 \cdot 10^{-3}$ at SNR = 0 dB which corresponds to a standard deviation of about 62.1 kHz = 12 ppm, down to $1.5 \cdot 10^{-5}$ at SNR = 20 dB, with a standard deviation of about 4.8 kHz = 0.9 ppm.

Downlink Burst. In this case the estimation range, as was shown in Table 2, is reduced. The c.i. is wider with respect to $\Delta f_{\max}$ than in the broadcast burst case, since more samples are used in the estimation. The maximum instability that can be correctly estimated extends up to ±0.13 MHz = ±25 ppm at SNR = 0 dB, and is raised to ±0.15 MHz = ±29 ppm at SNR = 20 dB.

The statistics (14) and (16) imply a reliable prediction of the carrier frequency offset estimation conditional mean and variance at SNR $\geq$ 10 dB, as actually required by the hypothesis that led to their writing. The discrepancy between the theoretical and simulation results at low signal to noise ratio, which is particularly evident at SNR $<$ 5 dB, derives from the heavy simplification introduced in Section 4.2.

5.2 Performance of Symbol Timing Estimation with Time-Dispersive Channels

Simulations were carried out with $h(t)$ channel models A and C, the power delay profiles of which are depicted in Fig. 5.

Channel A has a delay spread shorter than the guard interval, the transmitted signal stream is therefore not affected by the ISI caused by the dispersion of the channel impulse response. On the other hand, channel C has a higher delay spread than the guard interval. As an example of the estimators performance the results for uplink bursts with short preamble will be presented.

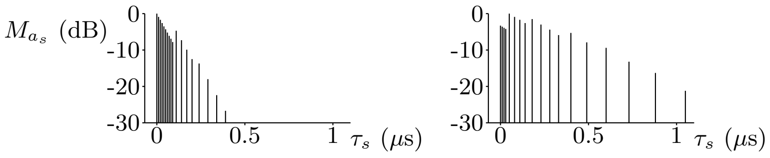


Fig. 5. Power delay profile of channel models A (left) and C (right)

We evaluate the symbol timing estimation performance by calculating the useful signal normalized power for each realization of the channel. The Cumulative Distribution Functions (CDFs) of $\overline{M}_{\mathrm{u}}$ and of the ratio $\overline{M}_{\mathrm{u}}/(\overline{M}_{\mathrm{int}} + M_w)$, with $\overline{M}_{\mathrm{int}}$ the normalized interference power and $M_w = 2\sigma^2$ the noise power are shown in Fig. 6. We observe that at SNR = 5 dB in about 95% of the total simulated transmissions with channel model A, the normalized useful power exceeds 0.9 and that in about only 5% of the realizations the signal to interference plus noise ratio is lowered by more than 0.5 dB with respect to the average channel SNR of 5 dB. We also see that in the case of channel C the corresponding CDFs show a performance loss, since in about 90% of the transmissions, the normalized useful power exceeds 0.9 and in about 15% the signal to interference plus noise ratio falls under 4.5 dB.

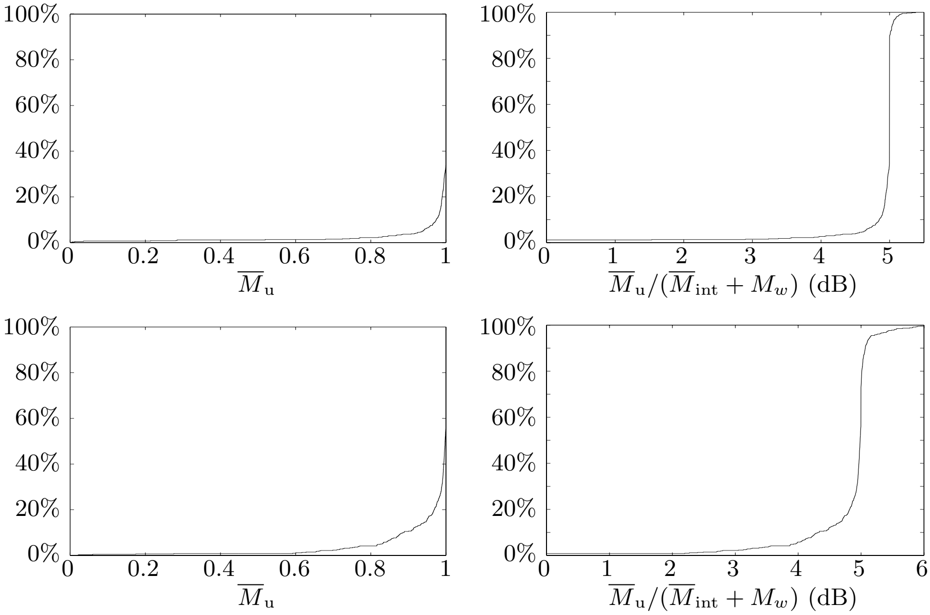


Fig. 6. Cumulative distribution functions of the normalized useful power (on the left) and of the signal to interference plus noise ratio (on the right) with channel models A (top) and C (bottom)

6 Conclusions

The main objects of this work are time and frequency synchronization issues for ETSI Hiperlan/2 standard. Algorithms for time and frequency offset estimation, which are present in literature, have been adapted to all the operation modes provided by a Hiperlan/2 system. Their efficiency has been tested through the simulation of burst transmissions in the presence of either an AWGN or time-dispersive channel impulse response.

A valuable technique for the evaluation of the performance of the symbol timing estimation in the presence of a time-dispersive channel has been proposed. It verifies the fraction of the useful power of the received signal after that the observation window has been placed on the OFDM symbol as indicated by the timing estimation algorithm.

An analytical description of the statistical properties of the carrier frequency offset estimation has been drawn under the hypothesis of high signal to noise ratio.

References

1. *Broadband Radio Access Networks; HIgh Performance Radio Local Area Network (Hiperlan) Type 2; Requirements and Architectures for Wireless Broadband Access*, ETSI Standard TR 101 031, ETSI, January 1999.
2. *Broadband Radio Access Networks; Hiperlan Type 2; System Overview*, ETSI Standard TR 101 683, ETSI, February 2000.
3. *Broadband Radio Access Networks; Hiperlan Type 2; Physical (PHY) Layer*, ETSI Standard TS 101 475, ETSI, February 2001.
4. L. Hanzo, W. Webb, T. Keller, *Single- and Multicarrier Quadrature Amplitude Modulation: principles and applications for Personal Communications WLAN and Broadcasting*, Wiley-IEEE Press, 2000.
5. T. Keller, L. Piazzo, P. Mandarini, L. Hanzo, "Orthogonal Frequency Division Multiplex Synchronization Techniques for Frequency-Selective Fading Channels", *IEEE Journal on Selected Areas in Communications*, vol. 19, no. 6, June 2001.
6. T.M. Schmidl, D.C. Cox, "Low-Overhead, Low Complexity [Burst] Synchronization for OFDM", *Proceedings of 1996 IEEE International Conference on Communications, ICC '96*, vol. 3, pp. 1301-1306.
7. T.M. Schmidl, D.C. Cox, "Robust Frequency and Timing Synchronization for OFDM", *IEEE Transactions on Communications*, vol. 45, no. 12, December 1997.
8. J. Medbo, P. Schramm, "Channel Models for Hiperlan/2 in Different Indoor Scenarios", ETSI EP BRAN 3ERIO85B, March 1998.
9. J. Medbo, "Radio Wave Propagation Characteristics at 5 GHz with Modeling Suggestions for Hiperlan/2", ETSI EP BRAN WG3 Temporary document XX, January 1998.
10. N. Laurenti, *Implementation Issues in OFDM Systems*, Ph.D. thesis, Università di Padova, February 1999.
11. N.M. Blachman "Gaussian Noise-Part II: Distribution of Phase Change of Narrow-Band Noise Plus Sinusoid", *IEEE Transactions on Information Theory*, 1988.

Performance Analysis of a Forwarding Scheme for Handoff in HAWAII

Chris Blondia[1], Olga Casals[2], Llorenç Cerdà[2], and Gert Willems[1]

[1] University of Antwerp, Dept. Mathematics and Computer Science,
Universiteitsplein 1, B-2610 Antwerpen, Belgium
{blondia, gewillem}@uia.ua.ac.be
http://win-www.uia.ac.be/u/pats/

[2] Polytechnic University of Catalunia, Computer Architecture Dept.,
Gran Capitan, D-6, E-08071 Barcelona, Spain
{olga,llorenc}@ac.upc.es
http://research.ac.upc.es/XARXES/CompNet/

Abstract. Demand for mobile network access is having a huge increase nowadays. Cellular networks are being deployed to cope with a high number of users. Several IP micro-mobility protocols have been proposed to handle routing and handoffs inside cellular networks. In this paper the HAWAII micro-mobility protocol is analyzed by means of an analytical model. A detailed description of the handoff procedure is given and illustrated by means of traces obtained from simulation. Several of the system details are taken into account in the analytical model. This allows us to investigate the influence of various system parameters (e.g. cell overlap area, beacon latency, forwarding buffer capacity, etc.) on the system performance for constant bit rate (UDP) traffic. The results are validated by means of simulation results obtained with the *network simulator* (ns).

1 Introduction

Handheld computing devices, such as palmtop computers are becoming the preferred platform for nowadays personal applications. With the evolution of these devices from having a limited communication support, typical point-to-point interfaces (PSTN modem or RS-232 cable), towards high-speed packet radio access interfaces, the demand for network access to mobile users will grow exponentially. As demands increase for wireless communication services, such as high-speed Internet access, video and high-quality image transmission, the wireless network access infrastructure will have to support a variety of applications and access speeds which should result in a service with the same level of quality as wireline users. Higher speed can be achieved in a cellular network by considering smaller cells. However, the smaller the cells are, the higher the frequency of handoffs may be. Mobile IP [1] (MIP), the current support of mobility in IP networks, delivers packets to a temporary address assigned to the mobile host at its current point of attachment. This temporary address is communicated to a possibly distant Home Agent. This approach applied to an environment with

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 503–514, 2002.

frequent handoffs may lead to high associated signalling load and unacceptable disturbance to ongoing sessions in terms of handoff latency and packet losses. Therefore, a hierarchical mobility management approach has been proposed (see e.g. [2]), where MIP supports wide area mobility (e.g. mobility between different operators) while local mobility is handled by more optimized micro-mobility protocols. These protocols should incorporate a number of important design features related to location management, routing and handoff schemes. They should fulfill requirements such as simplicity to implement, scalability with respect to the induced signalling, efficiency and performance with respect to packet loss and introduced delay. Prominent solutions for micro-mobility support are HAWAII ([5]) and Cellular IP ([4]).

In this paper, the performance of a handoff scheme based on packet forwarding used in HAWAII is evaluated by means of a simulation study and analytical modeling. The models that are presented allow to compute characteristic performance measures related to packet loss due to the expiration of the playout time and the delay experienced by packets involved in the handoff, together with the influence of various system characteristics on these performance measures. In [5], the expected number of dropped or lost packets as a function of the playout time has already been investigated. The aim of this study was to compare different path setup schemes with MIP and MIP with Route Optimization. Contrary to our paper, [5] does not investigate the influence of system details such as cell overlap area, beacon signal latency, time out and capacity of the forwarding buffer, on the system performance.

The remainder of this paper is structured as follows. In section 2, the HAWAII protocol and the forwarding scheme for handoff is explained. A detailed description of the protocol as implemented in the ns simulator is made in Section 3. In Section 4 a simple analytical model to assess the packet loss and the experienced packet delay due to handoff is presented. Section 5 is devoted to numerical results, showing the influence of the different system parameters on the performance using the analytical model. A model validation is made based on simulation results. Finally section 6 concludes the paper.

2 A Forwarding Scheme for Handoff in HAWAII

HAWAII is a domain-based approach for supporting mobility ([5] and [3]). The gateway into each domain is called the domain root router. The Mobile Host (MH) keeps it network address unchanged while moving within a domain. The Corresponding Hosts (CH) and the Home Agent (HA) do not need to be aware of the host's position within the domain. To reach the MH, HAWAII uses specialized path setup schemes that update forwarding entries in specific routers. When a router receives a packet for an unknown MH, it uses a preconfigured default interface pointing towards the domain root router. The packet will be forwarded in that direction till it arrives at a router knowing a route to the MH. There are two classes of path setup schemes for updating routing information: one for networks with MHs that can only maintain connection to one base station (e.g.

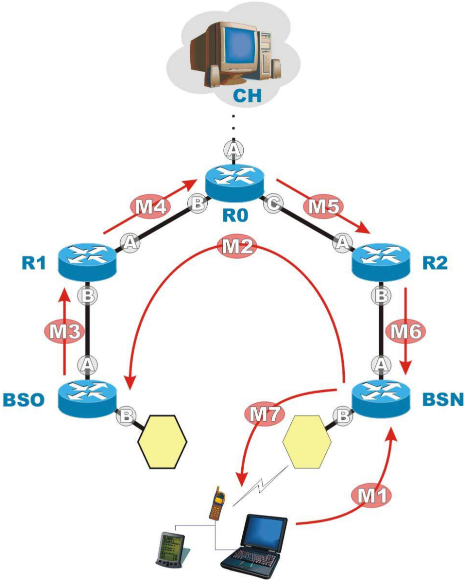


Fig. 1. Messages in MSF.

TDMA networks) and the other one for networks with MHs that can be connected to two or more base stations simultaneously like in CDMA networks for example. The first class includes two forwarding path setup schemes: the Multiple Stream Forwarding (MSF) and an alternative, the Single Stream Forwarding (SSF) scheme. These schemes forward packets from the Old Base Station (BSO) to the New Base Station (BSN) before being diverted at the crossover router (i.e. a router where the path from CH to BSO and the path from BSO to BSN cross). The second class includes two non-forwarding schemes: the Unicast Non-Forwarding (UNF) scheme and the Multicast Non-Forwarding (MNF) scheme. In these last schemes the BSO does not forward any packets to BSN. In this paper we limit our analysis to the MSF scheme although a similar analytical model can be applied to evaluate other schemes [6]. For the description of the MSF handoff protocol and its performance evaluation, we will use the following reference network (see Fig. 1). Let a MH move from the cell controlled by BSO to the cell controlled by BSN. The corresponding cells have a non-empty overlap area. Packets originating from the Corresponding Host (CH) reach BSO (resp. BSN) via the crossover router R0 and the intermediate router R1 (resp. R2).When the handoff is initiated, the MH closes the connection with BSO, establishes a connection to BSN and sends a MIP registration message (M1) to BSN, which in turn sends a path setup update message M2 to the BSO. All remaining packets arriving at BSO are stored in a buffer and forwarded to BSN when M2 arrives. Furthermore, when M2 arrives, BSO sends the path setup message (M3) to R1, who adds a forwarding entry to its routing table indicating that packets for the MH should leave the R1 via interface A. R1 sends the path setup message (M4) to R0, who adds a forwarding entry indicating that packets for the MH should leave the R0 via interface C. From this instant on, all packets arriving at router R0 are sent directly to BSN. The path setup message continues (M5 and M6) triggering similar actions until it reaches BSN. Remark that MSF can create transient routing loops (for example after BSO has changed its entry to forward packets but before R1 processes M3).

3 Detailed Description of the Handoff Procedure in the MSF Scheme

In this section we give a detailed description of a possible implementation of the MSF handoff procedure that corresponds to the one programmed in the network simulator (ns) used for the simulation analysis. We show the system parameters that may have a major impact on the handoff performance. These parameters will be considered in the performance evaluation.

All BS generate beacon signals at regular instants. The MH connects to the BS with the strongest beacon signal power. This implies that if a MH, connected to BSO, moves towards BSN it will initiate a handoff and connect to this Base Station after receiving the first beacon signal which was generated by BSN after the MH passed the middle of the overlap area of the cells controlled by BSO and BSN. Remark that as long as the MH is in the overlap area and the beacon of BSN is not generated, the MH stays connected to BSO and thus can still receive packets from BSO. Remember from Section 2 that BSN sends a path setup update message (M2) to BSO. BSO continues to send the packets to the MH, until M2 arrives at BSO. These packets will reach the MH only if it is inside the coverage of BSO. BSO uses a circular buffer of size FB packets to store the packets to be forwarded to BSN. We refer to this buffer as the *forwarding buffer*. All packets addressed to the MH are stored in this buffer (even after being transmitted to the MH). This allows packets, sent to the MH but lost because the MH moved out of coverage, to reach the MH when forwarded to BSN. Furthermore, the forwarding buffer is provided with a time out mechanism that ensures that a packet is held by the buffer only for a limited time period. When the path setup update message (M2) arrives at BSO, all packets outstanding in the buffer for which the time out is not expired are forwarded to BSN. Note that some of these packets may already have been successfully delivered to the MH.

Figure 2 illustrates the handoff procedure. The trace depicted in the figure has been obtained using the values of one of the ns simulations described later in this paper. The x-axis corresponds to the instants upon which new packets arrive at the MH, and the y-axis to the end-to-end delay of each packet. The figure shows the following time instants: the moment that the MH reaches the middle of the overlapping area of the two cells (intercell crossing), the instant when the MH crosses the border of the cell controled by BSO (end of coverage), the time instant at which the beacon is sent by BSN, and the moment the forwarding message M2 sent by BSN reaches BSO. For each arrived packet, the figure also indicates if it was sent by BSO, BSN or forwarded. The forwarded packets are those that are

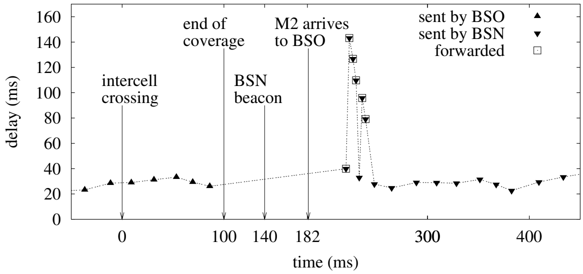


Fig. 2. Handoff procedure in the MSF.

turned around by some router (BSO or any intermediate router between BSO and the crossover router R0) because of the handoff. In this simulation a FB size of 10 packets is used with a time out of 100 ms. When the M2 message arrives at BSO, 6 packets are forwarded from BSO: the first one had already arrived at the MH (and thus is a duplicated packet not shown as forwarded in the figure). The other five are received by BSO between the end of coverage instant and the M2 message arrival instant. These packets are able to reach the MH after being forwarded to BSN and hence have a higher delay. Finally, note that the first forwarded packet arrives at BSN with a lower delay than the others. This is because this packet was turned around by a router located before BSO (this packet arrived at this router after the router was being updated with the M3 message sent by BSO).

4 An Analytical Model of the Forwarding Schemes

We introduce the following notations. Let $Rx(X)$ be the random variable denoting the time needed for a packet to be processed by router Rx and leaving via interface X. In other words, it denotes the time between the arrival of a packet at the router and its departure through the output interface X. (Same notation applies to "routers" BSO and BSN). Let (Rx, Ry) denote the propagation time on the link between router Rx and router Ry. Furthermore, we denote by $(Rx \to Rz^{\sigma})$ the time needed for a packet (or message) to travel from Rx to Rz. The superscript $^{\sigma}$ indicates that the processing of the specific router is included. Assume that the time (in ms) the MH needs to cross the overlap area is given by o ms. Finally, let τ_b denote the beacon latency, i.e. the difference between the time instant the beacon signal is generated and the instant the MH reaches the end of BSO's coverage area.

We now present a mathematical model for the MSF scheme. The following assumption is essential for computational tractability reasons. All routers involved in the path setup scheme are modeled as simple $M/M/1$ queues. The exponential service time of a packet includes both the processing time and the transmission time. Denote the load of a router by ρ and the exponential service rate by μ. Hence, the random variable $Rx(X)$, being the response time in an

$M/M/1$ queue, is exponentially distributed with rate $\mu(1-\rho)$. With this assumption, $(Rx \rightarrow Rz^{\sigma})$ is the sum of exponential variables (e.g. $Rx(X)$) and fixed propagation delays (e.g. (Rx, Ry)).
The packets involved in a handoff can be divided in classes according to the path they follow. The timing of these classes is given from the point of view of the arrival of a packet at router R0. Denote t_m as the instant the MH passes the middle of the overlap area, or the instant the MH crosses the border between 2 cells when there is no overlap. We consider all those packets that arrive at BSO after this instant as involved in the handoff and divide them in 4 classes.
- *Class 1*: These packets are processed by the BSO after t_m and before the new forwarding entry is added in the tables of the BSO. They arrive at R0 after
$t_0 = t_m - (R0^{\sigma} \rightarrow BSO^{\sigma})$
but before
$t_1 = t_m + \frac{o}{2} + \tau_b + (MH \rightarrow R0^{\sigma})$.
Class 1 packets are either directly sent to the MH, or they will eventually be forwarded to the BSN from the FB, unless they are removed from this buffer due to time out expiration or buffer overflow.
- *Class 2*: These packets arrive at R1 before M3 causes the adding of the new forwarding entry and they arrive at the BSO after its forwarding table has been updated by M2. Those packets are therefore directly forwarded from BSO to BSN without being put in the forwarding buffer. At time instant
$t'_1 = t_1 + (R0 \rightarrow BSO \rightarrow R1^{\sigma})$
M3 is processed and router R1 changes its forwarding entries for packets with destination the MH. Hence, packets arriving at R0 in the interval $[t_1, t_2]$ with
$t_2 = t'_1 - (R0^{\sigma} \rightarrow R1)$
belong to class 2.
- *Class 3*: These packets arrive at R0 before M4 causes the adding of the new forwarding entry at R0, and they arrive at R1 after t'_1. At time instant
$t_3 = t_2 + (R0^{\sigma} \rightarrow R1) + (R1 \rightarrow R0^{\sigma})$
router R0 changes its forwarding entries for packets with destination the MH. Therefore class 3 packets are those arriving at R0 in $[t_2, t_3]$.
- *Class 4*: These packets arrive at R0 after time instant t_3 and are directly diverted towards BSN. They experience no extra delay due to the handoff.

Consider a constant bit rate stream of packets originating from the CH, arriving at router R0 with a constant interarrival time of T ms. We let the arrival instant u of the first packet be uniformly distributed over $[t_0, t_0+T]$ and set $t_0 = 0$. Then it is possible to compute the probability distribution of the end-to-end delay for each packet traveling from the CH to the MH. Consider the k-th packet, arriving at R0 at time instant $(k-1)T+u$, and denote its end-to-end delay by $(\text{e-e})_k$. The probability that $(\text{e-e})_k$ is larger than t, depends basically on the class it belongs to and is given by the following expression.

$$
\begin{aligned}
&\mathrm{P}[(\text{e-e})_k > t] = \\
&\mathrm{P}[(k-1)T+u < t_1]\ \mathrm{P}[(\text{e-e})_k > t \mid (k-1)T+u < t_1] \\
&+ \mathrm{P}[t_1 < (k-1)T+u < t_2]\ \mathrm{P}[(\text{e-e})_k > t \mid t_1 < (k-1)T+u < t_2] \\
&+ \mathrm{P}[t_2 < (k-1)T+u < t_3]\ \mathrm{P}[(\text{e-e})_k > t \mid t_2 < (k-1)T+u < t_3] \\
&+ \mathrm{P}[t_3 < (k-1)T+u]\ \mathrm{P}[(\text{e-e})_k > t \mid t_3 < (k-1)T+u]
\end{aligned}
$$

For class 2, 3 and 4 respectively, we obtain

$$\mathrm{P}[(\text{e-e})_k > t \mid t_2 < (k-1)T + u < t_2] = \mathrm{P}[(CH \to BSO \to R0 \to BSN^\sigma) > t]$$

$$\mathrm{P}[(\text{e-e})_k > t \mid t_2 < (k-1)T + u < t_3] = \mathrm{P}[(CH \to R1 \to R0 \to BSN^\sigma) > t]$$

$$\mathrm{P}[(\text{e-e})_k > t \mid t_3 < (k-1)T + u] = \mathrm{P}[(CH \to R0 \to BSN^\sigma) > t].$$

The probability for a class 1 packet depends on k and is quite complicated as this involves the length of the overlap area, the instant the beacon is sent, the time out in the forwarding buffer and the size of this buffer. We distinguish three different cases for a class 1 packet:

- case 1: the k-th packet is directly sent from BSO to MH.
- case 2: the k-th packet is forwarded to MH via BSN.
- case 3: the k-th packet is lost at BSO (due to time out or buffer overflow).

For case 1 we obtain

$$(\text{e-e})_k = (CH \to R0 \to BSO^\sigma),$$

for case 2

$$(\text{e-e})_k = (CH \to R0 \to BSO^\sigma) + (t_1 - (k-1)T - u) + (BSO^\sigma \to R0 \to BSN^\sigma)$$

and for case 3 the end-to-end delay can be considered to be infinite. In case 2, we consider a possible loop between BSO and R1 for the first few packets that are forwarded from the FB, and we account for possible extra delay due to the burst of packets that is created when the FB is emptied. The details are omitted in this paper. We also need to determine the probability that the k-th packet finds itself in case 1, 2 or 3 respectively. Again most details are omitted and case 3 is considered as an example. Clearly,

P[k-th packet is in case 3] = P[(k-th packet is processed by BSO after t_d) AND (k-th packet is timed out OR pushed out of the FB)],

where t_d is the instant of disconnection of the MH from the BSO, directly depending on the values of o and τ_b. This probability yields a rather complex expression in which the probability that a packet is pushed out of the circular buffer FB before it could be forwarded is required:

P[k-th packet is pushed out] = P[(k+buffersize-1)$T + u < t_1$]

which is the probability that the (k+buffersize)-th packet needs to be forwarded (and therefore pushes out the k-th packet). The probablity that a packet is timed out is also required:

P[k-th packet is timed out] = $\mathrm{P}[t_{p_k} < t_1 + (R0 \to BSO^\sigma) - TO]$

where t_{p_k} denotes the instant of the end of the processing of the k-th packet at BSO and TO is the time out value. Remark that $t_1 + (R0 \to BSO^\sigma)$ equals the instant that the FB is emptied and the buffered packets are forwarded.

It is clear that, due to the $M/M/1$ assumption, all probabilities that occur in the above formulas can be computed through some standard conditional probability techniques. Similarly, the model allows us to compute several performance measures such as the expected number of packets arriving late at a playout buffer in the MH, due to the extra delay introduced by the forwarding scheme.

5 Performance Evaluation of the MSF Scheme

In this section we consider three cases in order of increasing complexity. In the first case (Section 5.1), cells do not overlap ($o = 0$ ms) and the beacon signal is received by the MH at the moment that it crosses the border between the cell controlled by BSO and the cell controlled by BSN. The time out (TO) and the capacity of the forwarding buffer in BSO (FB) are supposed to be large enough so that they do not cause packet loss. In the second case (Section 5.2) the cells overlap ($o > 0$ ms). The first beacon sent after the MH crosses the middle of the overlap area (determining the handoff instant) may occur while the MH is in the overlap area or after the MH has left the overlap area. Again in this case, no packets will be lost due to time out and/or forwarding buffer overflow. In the third case (Section 5.3) we have the same characteristics as the second one, except that the time out value and the forwarding buffer capacity are chosen so that packets may be lost in the forwarding buffer. In all three cases we consider

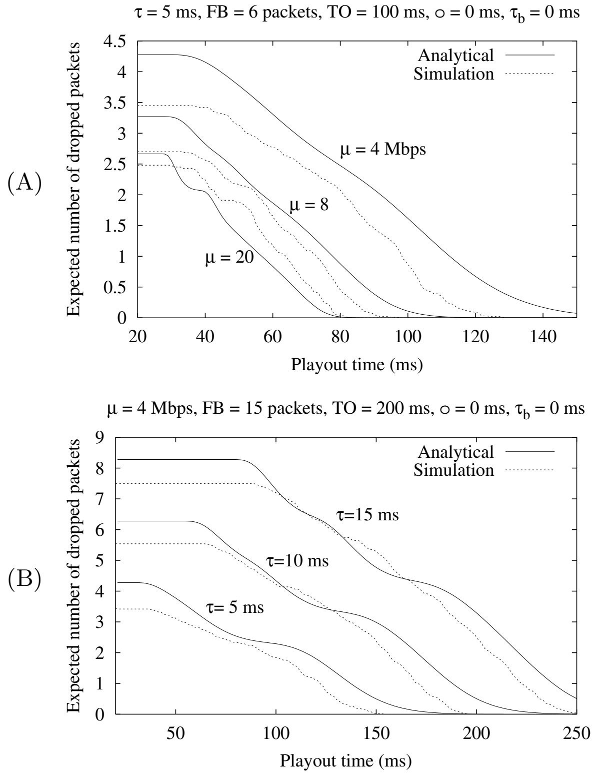


Fig. 3. Expected number of dropped packets vs. playout time for variable transmission rate μ (A) and for variable link delays (B).

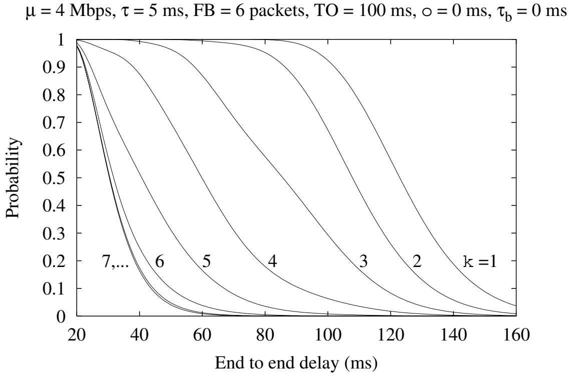


Fig. 4. Delay Distribution of k-th Packet.

the system shown in Fig. 1 where: (i) the fixed propagation delay between routers (Rx, Ry) is the same for all routers (also between CH and R0) and we will refer to it as τ, (ii) the correspondent node (CH) transmits 500 byte packets every 20 ms to the MH, (iii) the background traffic is modelled as Poisson sources such that each router has a load ρ of 0.8.

5.1 Delay Evaluation

The playout time is the maximum allowed end-to-end delay: if a packet's end-to-end delay exceeds this playout time, it will be dropped. In Fig. 3 the expected number of forwarded packets that are dropped due to expiration of playout time is shown as a function of the playout time, for different values of the transmission rate μ in the routers in Fig. 3.A and for different values of distance τ between neighbouring routers in Fig. 3.B .
The analytical results are compared against simulation results. The difference between simulation and analytical results is due to the $M/M/1$ approximation in the analytical model resulting in exponential packet service times, while in the simulation packets have constant length.

Fig. 4 depicts the distribution of the end-to-end delay of the k-th packet involved in the handoff when there is no overlap between the cells. As can be expected, the delay decreases with increasing sequence number of the packet. Starting from packet 7, the curves converge since the probability to experience some extra delay due to forwarding decreases rapidly.

5.2 Influence of the Beacon Latency

Fig. 5 shows the important influence the beacon latency has on the system performance. The expected number of forwarded packets dropped due to expiration of playout time is shown as a function of the playout time, for different values of the time between the instant the MH crosses the coverage area of BSO and the

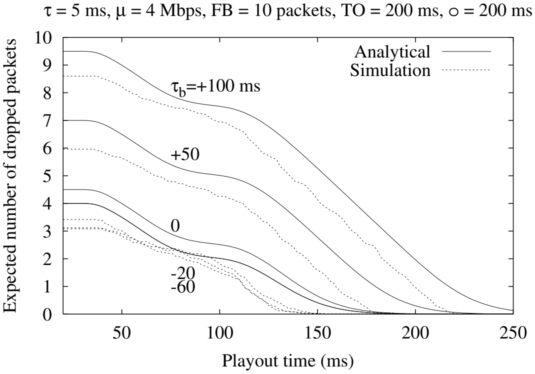


Fig. 5. Expected number of dropped packets vs. playout time for variable beacon latency τ_b.

instant that the first beacon originating from the BSN is received. A positive value of τ_b means that the beacon arrives after the end of the coverage area of BSO, while a negative value means that the beacon arrives before the end of the coverage area of BSO. Again the analytical results are validated with the ns simulation. When the beacon arrives after the end of the coverage area of BSO a higher number of packets will have to be stored in the forwarding buffer of the BSO which have to be forwarded to BSN once the path setup message triggered by the beacon arrival is received in BSO. This is shown in Fig. 2 for a beacon latency of 40 ms. This forwarding increases the end-to-end delay and therefore a higher number of packets will be dropped for a given playout time. Fig. 6 shows the delay distribution of the packets involved in the handoff procedure (i.e. the packets directly sent to the MH or forwarded after the instant the MH crosses the middle of the overlap area). From this figure, it is clear that the first packet

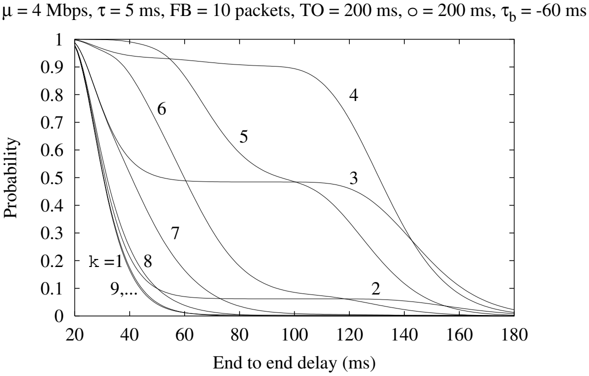


Fig. 6. Delay Distribution of k-th packet.

has a high probability to be sent directly to the MH without being forwarded via the BSN. Starting from packet 2, the probability of being forwarded increases. While these packets have a high probability to belong to class 1, starting from packet 7, the probability to belong to class 2 or 3 increases. Packet 9 and the following packets have a high probability to be sent directly to the MH via R2 and BSN and therefore, their delay distribution is close to the one of packet 1.

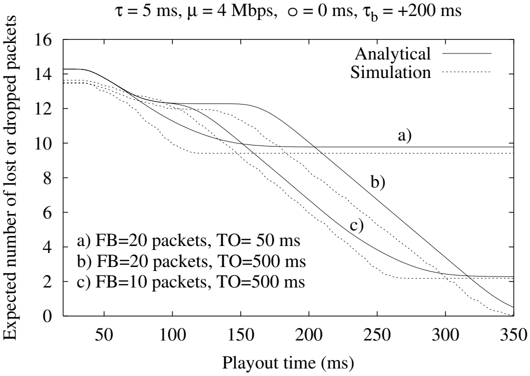


Fig. 7. Expected number of lost or dropped packets vs. playout time for variable FB and TO.

5.3 Influence of the BSO Time Out and Forwarding Buffer Capacity

Forwarded packets may be dropped due to the expiration of the playout time or they may be lost when the time out in the FB expires or when they are pushed out of the circular FB when it is full. The expected number of packets dropped or lost is shown in Fig. 7 as a function of the playout time, for different values of the time out and different values of the capacity of the FB buffer. There is no overlap here and the beacon latency is $\tau_b = 200$ ms.

In a) with $TO = 50$ ms and $FB = 20$ some packets (+/- 10 packets) stored in the forwarding buffer will time out before the path setup message M2 reaches BSO and thus they will be lost. In b) with $TO = 500$ ms and $FB = 20$ no packets are lost. In c) with $TO = 500$ ms and $FB = 10$ only a few packets (+/- 2 packets) will find the forwarding buffer full with packets to be forwarded when arriving at BSO and thus will push out the packets at the head of the queue which will be lost.

6 Conclusions

In this paper, the MSF-HAWAII handoff protocol is analyzed by means of an analytical model. Its performance for constant bit rate real-time (UDP) traffic is characterized by two measures: the expected number of forwarded packets that are dropped due to the expiration of the playout time together with the expected number of packets lost in the forwarding buffer and secondly, the individual end-to-end delay distributions of the packets that are involved in the handoff. The model includes a number of system implementation characteristics that have a major impact on the system performance: size of the overlap area between neighboring cells, frequency of beacon signal generation, size of forwarding buffer and time out value used in the forwarding buffer. The numerical results obtained with the analytical model have been validated with the ns simulator, showing the accuracy of the model.

Application of the model to a simple reference network shows that longer forwarding routes (due to longer distances between routers, slower routers or low capacity of transmission links) lead to a higher number of expected packets lost due to the expiration of the playout time and longer delays experienced by individual packets. Furthermore, it is shown that the expected number of lost packets may drastically increase when the beacon signal reaches the MH after it left the area covered by the old base station. The numerical examples also show that engineering accurately the forwarding buffer (both its time out value and its capacity) is an important, but difficult task, as unappropriate values of time out or buffer capacity may lead to a major performance degradation due to the loss of several packets.

Acknowledgments. This work was supported in part by the EU, under project IST-11591, MOEBIUS; The 2nd and 3th author were supported by the Ministry of Education of Spain, grant TIC-1998-115-C02-01

References

1. C. Perkins, ed., "IP Mobility Support", IETF RFC 2002, October 1996.
2. R. Caceres and V. Padmanabhan, "Fast and scalable handoffs for wireless networks", in Proc. ACM MOBICOM '96, pp. 56-66, 1996
3. R. Ramjee, T. La-Porta, S. Thuel, K. Varadhan, L. Salgarelli, "IP micro-mobility support using HAWAII", Internet draft, July 2000.
4. A. Valko, "Cellular IP - a new approach of Internet host mobility", ACM Computer Communication Reviews, January 1999
5. Ramjee, R., La Porta, T., Thuel, S., Varadhan, K., and Wang, S., HAWAII: a domain based approach for supporting mobility in wide-area wireless networks, Proceedings of International Conference on Network Protocols, ICNP'99.
6. Blondia, C., Casals, O., De Cleyn. P., and Willems, G., "Performance analysis of IP micro-mobility handoff protocols", Proceedings 7th Int. Workshop on Protocols for High Speed Networks, PfHSN 2002
7. A. Campbell, J. Gomez, C. Y Wan, S. Kim, Z. Turanyi, A. Valko, "Cellular IP", IETF draft (draft-ietf-mobileip-cellularip-00.txt), January 2000.
8. A. Campbell, J. Gomez, S. Kim, A. Valko, C.-Y. Wan and Z. Turanyi, "Design, implementation and evaluation of Cellular IP", IEEE Personal Communications, August 2000, pp.42-49

Evaluating the Performance of a Network Management Application Based on Mobile Agents

Marcelo G. Rubinstein[1], Otto Carlos Muniz Bandeira Duarte[2], and Guy Pujolle[3]

[1] Depto. de Eng. Eletrônica e Telecom., Universidade Estadual do Rio de Janeiro,
FEN, Rua São Francisco Xavier, 524, 20550-013, Rio de Janeiro RJ, Brazil,
[2] Grupo de Teleinformática e Automação, Universidade Federal do Rio de Janeiro,
COPPE/EE, CP 68504, 21945-970, Rio de Janeiro RJ, Brazil,
[3] Laboratoire LIP6-CNRS, Université Pierre et Marie Curie,
4, Place Jussieu, 75252, Paris Cedex 05, France

Abstract. This paper analyzes mobile agent performance in network management, comparing it with the client-server model used by the SNMP (Simple Network Management Protocol). Response time results show that the mobile agent performs better than the SNMP when the number of managed elements ranges between two limits determined by the number of messages that pass through a backbone and by the mobile agent size that grows with the variables collected on the network elements.

Keywords: Mobile agents, network management, and scalability

1 Introduction

Most network management systems use SNMP (Simple Network Management Protocol) [1] and CMIP (Common Management Information Protocol) [2] protocols, which are based on a centralized paradigm. These protocols use the client-server model, on which the management station acts as a client that provides a user interface to the network manager and interacts with agents, which are servers that manage remote access to local information stored in a Management Information Base (MIB).

Performance management is one of the management functional areas identified in OSI Systems Management and addresses the availability of management information, in order to be able to determine the network load [3]. This kind of management needs access to a large quantity of dynamic network information, which is collected by periodic polling.

The operations available to the management station for obtaining access to the MIB are very low-level. This fine grained client-server interaction, called micro-management, and the periodic polling generate an intense traffic that overloads the management station [4], resulting in scalability problems. Network

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 515–526, 2002.

management can be distributed and scaled by the use of mobile agents, which are programs that help users to perform tasks on the network, acting on behalf of these users. These agents move to the place where data are stored and select information the user wants; saving bandwidth, time, and money.

This paper analyzes the performance of mobile agents in network management, which is also being investigated by several researchers. Baldi et al. [4] evaluate the tradeoffs of mobile code design paradigms in network management applications by developing a quantitative model that provides the bandwidth used by traditional and mobile code design of management functionalities. Bohoris et al. [3] present a performance comparison between mobile agents, CORBA, and Java-RMI on the management of an ATM switch running an SNMP agent. Response time and bandwidth utilization results are presented for the transfer of an array of objects (fictitious data). Gavalas et al. present experimental implementation results for the transfer of an aggregation of multiple variables on a local network of a few nodes. They also describe applications that use mobile agents to acquire atomic snapshots of SNMP tables and to get objects, from SNMP tables, that meet specific criteria [5]. Sahai and Morin [6] perform measurements of bandwidth utilization of mobile agent and client-server applications on an Ethernet LAN of a few nodes. None of these papers concerns the problem of scalability of network management based on mobile agents on a complex network with a high number of nodes and similar in shape to the Internet. In this paper, we compare the scalability of the network management based on mobile agents against traditional SNMP through the analysis of simulation and implementation results. Two prototypes of an application that gathers MIB-II variables, one based on mobile agents and the other only based on the SNMP, have been created and tested on a LAN. By acquiring parameters related to the network management and to the agent infrastructure, new results are obtained on large topologies similar in shape to the Internet.

This paper is organized as follows. Section 2 describes main network management systems used nowadays. Section 3 presents the implemented prototypes and measurement results. Section 4 reports simulation results. At last, concluding remarks are presented in Section 5.

2 Network Management Systems

In the SNMP, operations available to the management station for accessing the MIB are very low-level. This interaction does not scale well because of the generation of intense traffic and computational overload on the management station [4].

Some steps towards decentralization have already been taken by IETF and ISO organizations. In event notification, SNMP agents notify the management station upon the occurrence of a few significant events. These agents use traps, i.e., messages sent without an explicit request from the management station, to decrease the intensive use of polling. ISO uses more complex agents that have higher processing capacity. In both the approaches, the agent is only responsible for the notification of an event.

A more decentralized approach is adopted in SNMPv2 [1] (SNMP version 2), on which there are multiple top-level management stations, called management servers. Each such server is responsible for managing agents, but it can delegate responsibility to an intermediate manager. This manager, also called proxy agent, plays the role of a manager in order to monitor and control the agents under its responsibility and also works as an agent to provide information and to accept control from a higher-level management server. Version 3 of the SNMP, SNMPv3 [1], incorporates a new security scheme to be used with SNMPv2 (preferred) or SNMPv1. SNMPv3 is not a stand-alone replacement for SNMPv1 or SNMPv2.

The RMON (Remote MONitoring) [7] uses network monitoring devices called monitors or probes to perform proactive LAN monitoring on local or remote segments. These probes provide information about links, connections among stations, traffic patterns, and status of network nodes. They also detect failures, misbehaviors, and identify complex events even when not in contact with the management station.

These proposals seem to reduce the traffic around the management station, but as the computational power of the network nodes is increasing, it is possible to delegate more complex management functions to nodes. Moreover, in order to satisfy the diverse needs of today's network, new network management systems that analyze data, take decisions, and take proactive measures to maintain the Quality of Service (QoS) of the network must be developed. Mobile agents seem to be a good alternative to satisfy these needs.

Main advantages that may justify mobile agent utilization in network management are: reduced cost by using a semantic compression, which filters and selects only relevant information; asynchronous processing that allows the decoupling from the home node; flexibility that permits the substitution of the behavior for management agents in real-time; and autonomy, since the agent can take decisions, performing a reactive management based on task delegation.

Since SNMPv2 is not as spread as SNMPv1 and network management based on SNMPv1 does not scale when size or complexity of the network increases, mobile agents can be used to increase network management scalability.

3 Implementation of a Management Application

We compare two different solutions for gathering MIB-II [8] variables on managed elements: a mobile agent-based one and one only based on the SNMP.

The Mole infrastructure [9] is used in the mobile agent implementation. This system provides the functionality for the agents to move, to communicate with each other, and to interact with the underlying computer system. Two different kinds of agents are provided: system agents and user agents. System agents are usually interface components to resources outside agent systems. They have more rights than non-system agents (e.g., only system agents can read or write to a file), but they are not able to migrate. User agents are agents that have a "foreigner status" at a location, which means that they are not allowed to do something outside the agent system as long as they can not convince a system agent to give them access to outside resources [9].

Mole uses TCP to transfer mobile agents, which are implemented in Java. A weak migration scheme is provided, where only the state related to data, which contains global and instantiated variables, is transfered. As a consequence, the programmer is responsible for encoding the agent's execution state, which includes local variables, parameters, and execution threads, in program variables. This migration scheme is implemented by using the object serialization of Java. After the calling of the migrateTo()-method by an agent thread, all threads belonging to the agent are suspended. The agent is serialized by creating a system-independent representation. This serialized version is sent to the target that reinstantiates the agent. A new thread is started and as soon as this thread assumes control of the agent, a message is sent to the source that finishes all threads belonging to the agent and removes it from the system.

Both the implemented prototypes, one with mobile agents and the other without, use the SNMP protocol to gather MIB-II variables. The AdventNet SNMP library [10] and the snmpd from package ucd-snmp are used on the prototypes. The AdventNet SNMP package contains APIs to facilitate the implementation of solutions and products for network management. Version 2.2 of the AdventNet SNMPv1 has been used. The daemon snmpd, which is included in the Linux Red Hat, is an SNMP agent that responds to SNMP request packets. The package versions used on this experiment have been the 3.5.3 (for machines running the Red Hat 5.2) and the 4.0.1 (for the Red Hat 6.x).

3.1 The Two Implemented Prototypes

The mobile agent implementation (Figure 1) consists of one mobile agent, which migrates to all network elements to be managed, one SNMP agent, which accesses the MIB-II variables, and one translator agent, which converts the mobile agent request into an SNMP request. The mobile agent migrates to a network element (arc 1 of Figure 1) and communicates by Remote Procedure Call (RPC) with the translator agent (arc 2). This translator agent sends a request (GetRequest PDU of SNMP) to the SNMP agent (arc 3) and obtains the response (arc 4) that is passed to the mobile agent (arc 5). Then, the mobile agent goes to the next element (arc 6) and restarts its execution. After finishing its task, which consists of visiting all network elements to be managed, the mobile agent returns to the management station (arc n).

In the implementation that is only based on the SNMP, we have used the traditional model of this protocol. The manager sends an SNMP packet to an SNMP agent that responds to this manager. The manager sends requests to all elements to be managed, one after the other, i.e., a new request is started after receiving the response from the previous one, until the last network element receives a request and sends the response to the manager. This manager has been implemented in Java directly over the Java Virtual Machine.

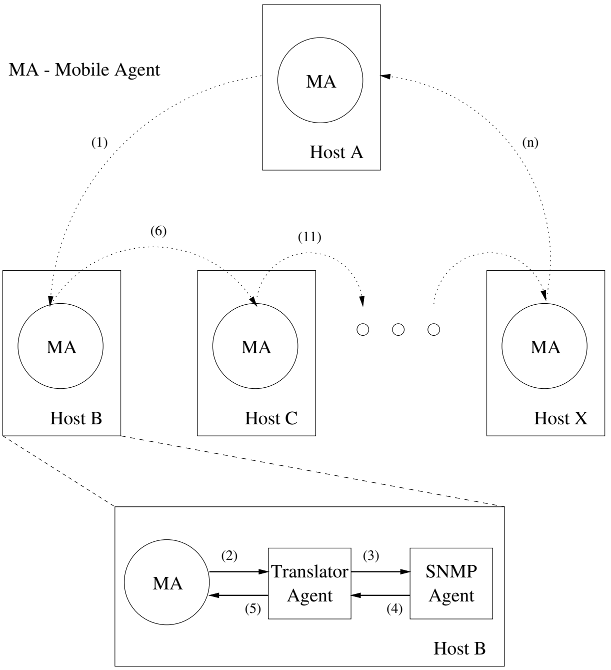


Fig. 1. Network management by using a mobile agent.

3.2 Experimental Study

We perform an experimental study in order to evaluate the scalability of the two implementations. The topology used on this experiment consists of one management station (host A) and two managed network elements (hosts B and C) interconnected through a 10 Mbps Ethernet LAN. Host A is a Pentium MMX 233 Mhz, with 128 Mbytes of memory and running Linux Red Hat 6.2. Hosts B and C are Pentiuns II 350 MHz, respectively with 64 Mbytes and 128 Mbytes of memory and running Linux Red Hat versions 6.1 and 5.2.

In order to evaluate the performance of the two prototypes for a great number of managed elements, we alternately repeat the two elements B and C, e.g., if we want 5 elements to be managed, we use an itinerary {B, C, B, C, B, and A}.

The considered performance parameter is response time in retrieving the MIB-II [8] variable *ifInErrors* from elements to be managed. This variable denotes the number of received packets discarded because of errors.

The JDK (*Java Development Kit*) 1.1.7 version 3 has been used. All measurements have been performed early in the morning or at night in order to limit the variations of network performance, which would influence the response time results. Both the implementations have been tested in the same conditions, using the same itinerary. We have made all the tests with the mobile agent platforms running uninterruptedly. The number of managed network elements has been varied from 1 to 250. For each measured parameter, 10 samples have been observed and we have calculated a 99 % confidence interval for mean. These intervals are represented in the figures by vertical bars.

The mobile agent carries with itself the name of the variable to be collected, the itinerary, and the already gotten responses. The SNMP sends a GetRequest PDU and receives a GetResponse PDU.

The effect of the number of managed elements in response time has been analyzed. In all figures, we present the sample mean.

Response time for the SNMP grows proportionally with the number of managed elements, since the time to manage a network element is approximately the same for all network elements (Figure 2). For the mobile agent, response time increases faster when the number of managed elements grows, due to the mobile agent size that grows with the collected variables on each network element. In the topology used on this experiment, the SNMP performs much better than the mobile agent.

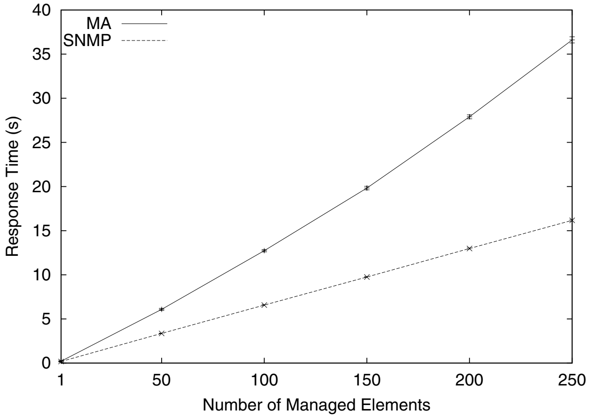


Fig. 2. Response time per number of managed elements.

Figure 3 presents time to access the MIBs and for the RPCs related to the communication between the mobile agent and translator agents. In this experiment and for the SNMP, for 250 managed elements, 99.6 % of the total time is spent on access to the MIBs. For the mobile agent, the access to the MIBs and the RPCs take 52.8 % of the total time for 250 managed elements. For the SNMP, the access to the MIBs grows proportionally with the number of managed elements and spends 65 ms per element. This MIBs access added to the RPCs related to the communication between the mobile agent and translator agents also grow linearly and spend approximately 78 ms per element.

The mobile agent remaining time is calculated by the difference between the total time for the mobile agent and the time for accessing the MIBs and

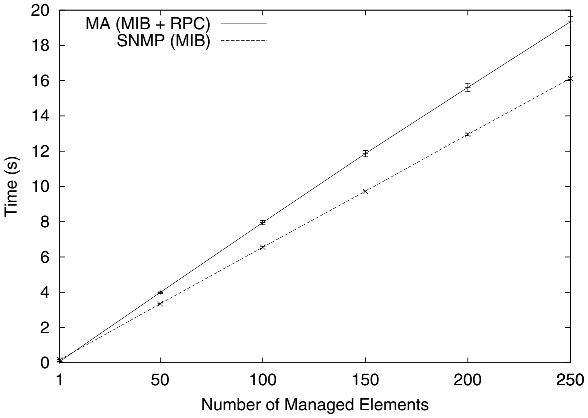


Fig. 3. Time to access the MIBs and for the RPCs.

for the RPCs (Figure 4). Since, for this experiment, agent transmission time is very small comparing to other times that constitute the total response time, the remaining time corresponds to infrastructure related times, e.g., serialization/deserialization, threads creation, and internal messages transmission.

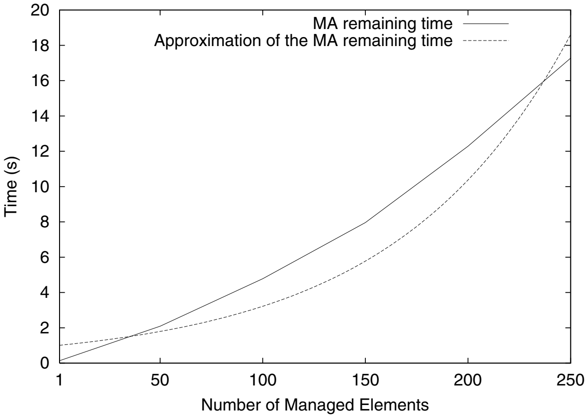


Fig. 4. Mobile agent remaining time.

The mobile agent remaining time grows exponentially with the number of managed elements, so the curve of the Figure 4 can be approximated to:

$$y = a^x, \text{ where } a = 1.01176 \ . \tag{1}$$

This approximation has been chosen to allow, in a simple way, its use in simulations assessed for more general topologies (Section 4).

4 Performance Analysis by Simulation

The applicability of mobile agents in carrying out network management tasks is also assessed by simulation.

The Network Simulator (NS) [11] is used in these simulations. This discrete-event simulator provides several implemented protocols and mechanisms to simulate computer networks with node and link abstractions. In these simulations, we have used the functionalities of Ethernet, topologies similar in shape to the Internet, and UDP and TCP protocols. Some UDP and TCP modules of the NS have had to be modified in order to allow the transmission of mobile agents.

The NS works with packets sent through a network and usually does not take into account processing time of the application layer on each node. For this reason, some parameters related to network management have been added to the simulation model. These parameters depend on the agent infrastructure, on the operational system, and on computer load, but their use turns simulation results more reliable to a real implementation. Table 1 contains the parameters used in the simulations.

Table 1. Parameters used in the simulations

Parameter	Value
Initial size of the agent	1500 bytes
Request size for *ifInErrors*	42 bytes
Response size for *ifInErrors*	51 bytes
MIB access time per node for the agent	78 ms
MIB access time per node for the SNMP	65 ms
Related to the remaining time for the agent	1.01176

The simulation model assumes that links and nodes have no load and that links are error-free. The Maximum Segmentation Size (MSS) used in the simulations is 1500 bytes, therefore, there is no fragmentation of SNMP messages since they are small. For the mobile agent, since the initial size is 1500 bytes, after visiting the first element, its size will be higher than the MSS, and so the agent will be fragmented and sent in different packets, damaging the performance. Every request of a variable is sent on a different message. In all simulations, the mobile agent follows a predetermined itinerary. The mobile agent uses the TCP-Reno as a transport protocol, because of its great use in the Internet, and the UDP protocol is used in the SNMP simulations.

Two kinds of topologies have been used in the simulations. The first type consists of elements in a 10 Mbps Ethernet LAN, with 250 nodes and latency of 10 μs. The second kind is similar in shape to the Internet. This topology is called transit-stub, because each routing domain in the Internet can be classified as either a stub domain or a transit domain [12]. A domain is a stub domain if the path connecting any two nodes u and v goes through that domain only if either u or v is in that domain. Transit domains do not have this restriction. The purpose of transit domains is to interconnect stub domains efficiently. A transit domain comprises a set of backbone nodes, which are typically fairly well connected to each other. In a transit domain, each backbone node also connects to a number of stub domains, via gateway nodes in the stubs.

These transit-stub topologies can be used in the network management of a matrix-branch organization, on which a matrix wants to manage their branches spread geographically. The management strategy used in this experiment for transit-stub topologies considers that the management station belongs to a node of a stub domain and managed network elements are located in other stub domains (Figure 5). In the matrix-branch case, the management station from the matrix manages the branch routers and each branch is represented by a stub and contains some routers.

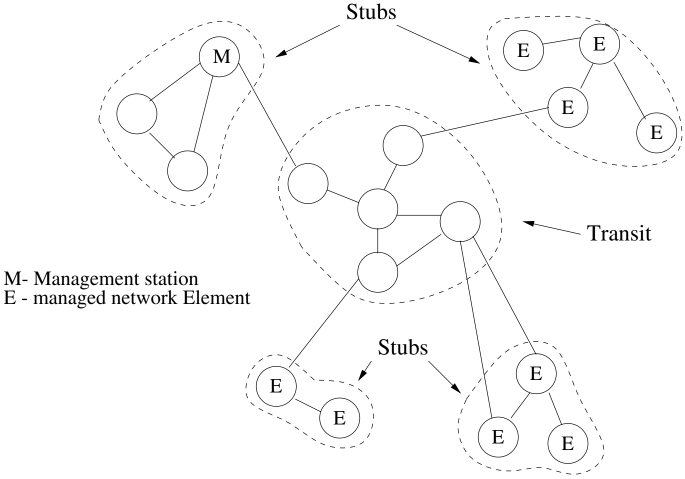


Fig. 5. Network management on a transit-stub topology.

The considered performance parameter is response time in retrieving the MIB-II variable *ifInErrors*.

We have used the LAN topology in order to compare the simulation model with the implementation results of Section 3.2.

Figure 6 presents response time for the mobile agent and for the SNMP, in implementation and simulation studies. We can say that the simulated models reproduce the behavior of the implementations. There is a little difference in the response time for the mobile agent due to the approximation of the remaining time that has been used in the simulations (Section 3.2).

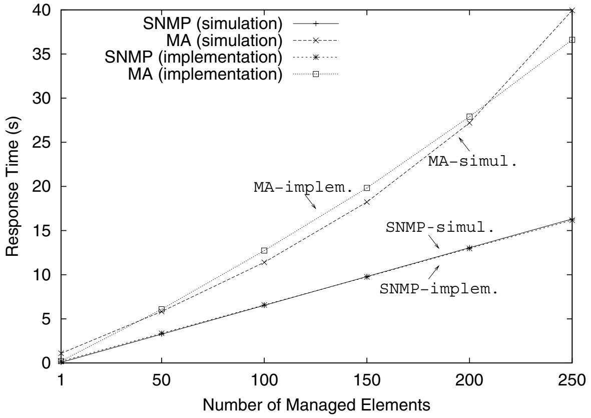


Fig. 6. Response time for implementation and simulation studies.

Mobile agent performance is also evaluated in a situation closer to the one found on the Internet, on which latencies are much greater than on LANs. Three different transit-stub topologies created by the topology generator GT-ITM [12] are used. The topologies have 272 nodes and links of these topologies have a 2 Mbps bandwidth and latency of a few milliseconds. The management station controls groups of 16 network elements, which is the number of nodes of a stub domain. Management is performed in a predetermined way: all elements of a stub are accessed, after that, the next stub is managed, until all the 16 stubs are accessed. If not specified, figures present mean response time for the three topologies.

Figure 7 shows that the mobile agent's behavior does not change with the topology, but for the SNMP, there is a little difference in response time for the three topologies. This variation is due to the great number of SNMP packets that traverse the backbone (transit) links and to the configuration of the backbone nodes that changes with the topology. Figure 7 also presents mean response time. For a small number of managed elements, the SNMP performs better than the mobile agent, due to the fact that the SNMP messages are smaller than the initial size of the mobile agent. As the number of managed elements increases, response time for the SNMP grows proportionally, since the time to manage a stub is approximately the same for all stubs. For the mobile agent, response time increases faster when the number of managed elements grows, due to the incremental size of the mobile agent. By extrapolating the analysis, we can conclude that the mobile agent performs better than the SNMP when the number of managed network elements ranges between two limits, an inferior and a superior one, respectively determined by the number of messages that pass through a backbone and the size of mobile agent that grows with the variables collected on network elements.

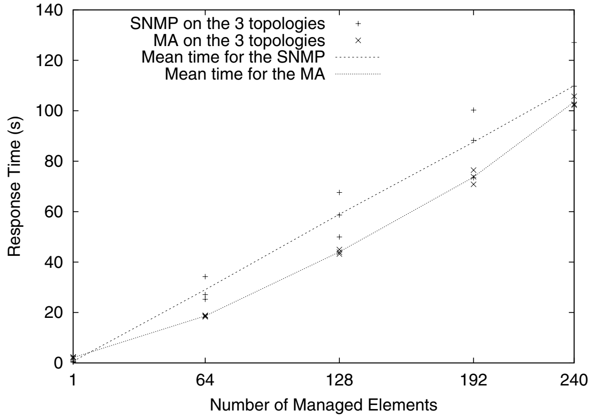


Fig. 7. Response time for the mobile agent and for the SNMP.

5 Conclusion

This work has analyzed the scalability of mobile agents in network management. The performance of mobile agents has been compared with the SNMP (Simple Network Management Protocol) one.

We have compared two prototype implementations for gathering MIB-II (Management Information Base - II) variables on managed elements: a mobile agent-based one and the pure SNMP. Results show that the mobile agents require a higher processing capacity and that the SNMP uses a larger number of messages related to the management station when the number of managed elements exceeds a value related to the overhead of several retrievals of GetRequest PDUs. The mobile agent infrastructure turns the execution of Java code slower, mainly because of serialization/deserialization, threads creation, and internal messages transmission. The topology used on the measurements is adverse to the mobile agent, since the great availability of bandwidth on the Ethernet turns message transmission times negligible comparing with processing times. Therefore, in this topology, the SNMP performs much better than the mobile agent.

Simulations of the two implementations have also been performed in the NS Network Simulator, in order to obtain results on large topologies similar in shape to the Internet. Response time results show that the mobile agent performs better than the SNMP when the number of managed elements ranges between two limits, an inferior and a superior one, respectively determined by the number of messages that pass through a backbone and by the mobile agent size that grows with the variables collected on network elements.

In a general way, we conclude that the mobile agent paradigm significantly improves the network management performance when subnetworks must be managed remotely; mainly if the links between the management station and the elements to be managed have a small bandwidth and a large latency.

Acknowledgements. This work has been supported by UFRJ, FUJB, CNPq, CAPES, COFECUB, and REENGE. We would like to thank FAPERJ for the grant to Mr. Rubinstein during the execution of this work on the Departamento de Engenharia Eletrônica e de Computação da UFRJ.

References

1. Stallings, W.: SNMP and SNMPv2: The infrastructure for network management. IEEE Communications Magazine **36** (1998) 37–43
2. Yemini, Y.: The OSI network management model. IEEE Communications Magazine **31** (1993) 20–29
3. Bohoris, C., Pavlou, G., Cruickshank, H.: Using mobile agents for network performance management. In: IEEE/IFIP Network Operations and Management Symposium (NOMS'00), Honolulu, Hawaii (2000) 637–652
4. Baldi, M., Picco, G.P.: Evaluating the tradeoffs of mobile code design paradigms in network management applications. In: 20th International Conference on Software Engineering (ICSE'98), Kyoto, Japan (1998) 146–155
5. Gavalas, D., Greenwood, D., Ghanbari, M., O'Mahony, M.: Mobile software agents for decentralised network and systems management. Microprocessors and Microsystems **25** (2001) 101–109
6. Sahai, A., Morin, C.: Towards distributed and dynamic network management. In: IEEE/IFIP Network Operations and Management Symposium (NOMS'98), New Orleans, USA (1998)
7. Waldbusser, S.: Remote network monitoring management information base. RFC 1757 (1995)
8. McCloghrie, K., Rose, M.: Management information base for network management of TCP/IP-based internets: MIB-II. RFC 1213 (1991)
9. Baumann, J., Hohl, F., Straber, M., Rothermel, K.: Mole - concepts of a mobile agent system. World Wide Web **1** (1998) 123–137
10. Advent Network Management Inc.: AdventNet SNMP release 2.0. http://www.adventnet.com (1998)
11. Fall, K., Varadhan, K.: NS Notes and Documentation. Technical report, The VINT Project (1999)
12. Zegura, E.W., Calvert, K.L., Donahoo, M.J.: A quantitative comparison of graph-based models for internet topology. IEEE/ACM Transactions on Networking **5** (1997) 770–783

Performance Evaluation on WAP and Internet Protocol over 3G Wireless Networks

Hidetoshi Ueno, Norihiro Ishikawa, Hideharu Suzuki, Hiromitsu Sumino, and Osamu Takahashi

NTT DoCoMo, Multimedia Laboratories
3-5, Hikari-no-oka, Yokosuka, Kanagawa, 239-8536, Japan
{hueno, ishikawa, hideharu, sumino, osamu}@mml.yrp.nttdocomo.co.jp

Abstract. This research analyses the performance of WAP 1.x in a comparison to the Internet protocol. We implement a WAP client and a WAP gateway based on WAP version 1.1 and assess the response time by comparing to that of HTTP and TCP. We use a W-CDMA simulator to evaluate its performance in high-speed wireless networks such as 2.5G and 3G. The results shows that both protocols have comparable performance (i.e. response time) except when transmitting large content sets (e.g. multimedia data files), in which case the performance of HTTP/TCP is better than that of WAP 1.x. We also evaluate WAP specific functions such as the binary encoding of WAP headers and contents. While binary encoding is effective for small content sets, its effectiveness and performance are questionable for large content sets. Finally, we propose a mobile Internet architecture that is suitable for 2.5G and 3G wireless networks based on the evaluation and our experience with the i-mode service. Our architecture consists of wireless optimized TCP, TLS, HTTP and XHTML.

1 Introduction

Services that will access the Internet from handheld devices such as weather forecasts, news, and mobile banking are attracting people's attention. Handheld devices tend to have many restrictions such as have less powerful CPUs, less memory, and smaller displays. Wireless networks also suffer from higher error rates, lower bandwidth, higher latency, and unexpected circuit failures. Since it was considered that the protocol used in the Internet might not be suitable for wireless environments, the Wireless Application Protocol (WAP) Forum developed the WAP version 1.x (WAP 1.x) protocol [1].

WAP 1.x is designed for various kinds of wireless network bearers (e.g. GSM and CDMA) [1]. However, it is unclear what sort of networks can take full advantage of WAP 1.x, and its performance of has not been fully evaluated. Thus, we developed a WAP client and a WAP gateway based on the WAP specifications [1], and then evaluated WAP performance by using a Wideband Code Division Multiple Access (W-CDMA) simulator. We compared the WAP 1.x protocol to the Internet protocol (HTTP/TCP). Finally, we created a mobile Internet architecture that is suitable for

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 527-538, 2002.

next generation (2.5G) and third generation (3G) wireless networks since services over the International Mobile Telecommunication 2000 (IMT-2000) networks [2] have been started in Japan.

The WAP 1.x protocol is overviewed in section 2, and its implementation in the WAP 1.x test-bed system is described in section 3. Our evaluation of WML 1.x binary encoding is given in section 4 and is compared to WAP 1.x and HTTP/TCP in section 5. In section 6, we propose a mobile Internet architecture for high-speed wireless networks such as 2.5G and 3G.

2 WAP 1.x Overview

WAP defines an architecture and protocols with the goal of providing special functionalities such as telephony, push delivery, and suspend & resume. The following summarizes the WAP architecture and protocols.

The WAP architecture consists of WAP client, WAP getaway and origin server. It is based on the Internet World Wide Web (WWW) model with few enhancements. The WAP protocols are used between the WAP client and the WAP gateway, and the Internet protocols (i.e., HTTP and TCP) are used between the WAP gateway and the origin server. Optimizations and extensions have been made in order to satisfy the requirements of the wireless environment.

The WAP protocols consist of Wireless Datagram Protocol (WDP), Wireless Transport Layer Security (WTLS), Wireless Transaction Protocol (WTP), Wireless Session Protocol (WSP) and Wireless Application Environment (WAE) (Figure 1.).

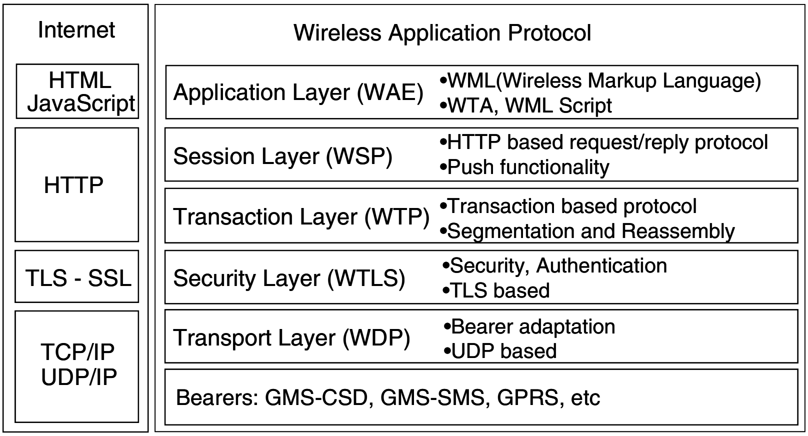


Fig. 1. WAP protocol overview and its comparison to the Internet protocol

WDP provides functions for bearer adaptation, which absorbs the differences of lower wireless network protocols. When the bearer network supports IP, UDP is used for WDP. WTLS is developed based on TLS and provides the means for supporting security functions such as authentication and confidentiality. WTP provides transaction type communication and has three transaction types (i.e. class 0, 1, and 2).

Class 2 transaction in particular realizes reliable communication by supporting packet retransmission. WTP supports segmentation and reassembly (SAR), which provides the means for transmitting large content sets whose size exceeds one Maximum Transfer Unit (MTU). WSP provides session management functions including session initiation and session suspend and resume. WSP provides header compact encoding and push capability in addition to the basic functionality of HTTP.

WAE is a general term of application environments in WAP, and consists of several components. WAP defines the Wireless Markup Language (WML) [1] as the markup language, and WML Script as the scripting language, and Wireless Telephony Application (WTA) as the telephony-related application. WML is based on Extensible Markup Language (XML) and defines its own tags, and has no compatibility with Hyper Text Markup Language (HTML). WML uses the binary representation format [1] in order to reduce content transfer volume. WML content is encoded into binary representation format at the WAP gateway to the wireless network.

2.1 WAP Related Works

Since there are a lot of Internet contents written in HTML, WAP clients must be able to access HTML contents. To do so the WAP gateway needs to support content conversion from HTML to WML. Reference [3] investigated the problems associated with the conversion from HTML to WML. It proposed some techniques for converting HTML to WML but several problems remained. The overhead of content conversion is also an issue because poor gateway scalability becomes a serious handicap when the number of subscribers increases. Although it is very important to investigate and consider WAP gateway scalability, research has been insufficient to date.

As for evaluating the WAP protocol, one paper evaluated the WTP class two protocols by implementing the WAP protocol stack [4]. It pointed out some inconsistencies in the WAP specifications but didn’t investigate WTP performance over wireless networks.

Reference [5] analyzed the network traces generated by a mobile browser application. The research observed daily and weekly cycles, and found some evidence of self-similarity in the network traffic produced by the application. The research also compared and contrasted the mobile browser traffic characteristics with the results for WWW traffic published in the literature. The results of this research are very significant for the design of wireless networks. However, since the network characteristics of 3G networks are quite different from those of 1G and 2G networks, in-depth research of 3G network traffic characteristics is needed.

While 3G commercial services started in Japan in October 2001, no research has examined WAP 1.x over 3G networks.

3 Implementation of WAP 1.x Client and Gateway

We developed a WAP client and a WAP gateway based on WAP version 1.1 (WAP 1.1) specifications [1], and simulated a high-speed wireless environment by using a hardware-based W-CDMA emulator. The W-CDMA emulator allows several of the parameters related to the wireless network (e.g. bearer speed, error rate and maximum number of retransmission) to be set up. The parameters used, see Table 1, are based on the FOMA implementation[1].

Note that WAP 1.1 is not the newest version of the WAP 1.x series, but there is no measurable difference as regards protocol performance.

Table 1. This table shows principal parameters set in the W-CDMA bearer simulator. In W-CDMA, one to twelve PDUs (i.e. Radio Link Control frames) constitute one Forward Error Correction (FEC) frame [6]. The actual size of the FEC frame depends on the link conditions and bandwidth allocation. Since the error rate value is based on the typical average value on our experimental 3G system, it captures wireless-specific characteristics (e.g. fading behavior). The error rate is per FEC frame.

Parameter	Value
Bearer Speed	(Downlink) 384 and 64 Kbps, (Uplink) 64 Kbps
Layer 2	Radio Link Control (RLC) Protocol [6]
Bearer MTU	1500 bytes
Error Rate	5% (per FEC frame)

3.1 WAP Test-Bed System Overview

The WAP 1.1 test-bed system is shown in Figure 2.

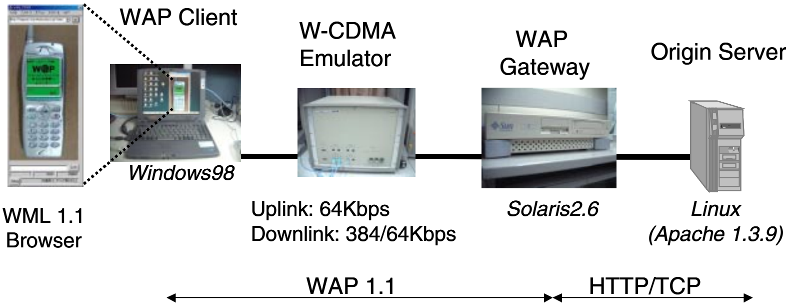


Fig. 2. This is the WAP 1.1 test-bed system. The W-CDMA emulator is set between the WAP client and the WAP gateway.

1) WAP client

We wrote the WAP client in C++ for the Windows 98 platform. It consists of WML 1.1 browser, WAP protocol module and so on. The WML 1.1 browser can emulate a mobile client and has the capability of displaying WML contents.

2) WAP gateway

We wrote the WAP gateway in C language for Solaris 2.6. It consists of WAP and Internet protocol modules, gateway application module, and so on. The

[1] FOMA is the first 3G commercial service; it started in October 2001. For further information, please see (http://www.nttdocomo.co.jp/english/).

gateway application module realizes WML content pursing and binary encoding if the content is WML, and forwards the encoded data to the WAP client.

3) Origin Server

We used Apache 1.3.9 as the WWW server application. It also supports the Common Gateway Interface (CGI) function so that it is possible to create dynamic WML content.

3.2 WAP Applications

Table 2 lists the applications we implemented in the test-bed system. GET and POST are used for Web browsing. As for the push application, the WAP gateway becomes a push server and pushes contents to the WAP client by using the WSP push or confirmed push method. We also developed an e-mail application by using CGI and an e-mail receiving application by using WSP push.

Table 2. Applications developed for the test-bed system

Service	Content	WSP Function
Browsing	WML	GET, POST method
Push	GIF, Text	WSP Confirmed/Unconfirmed Push
E-mail	Text	(Send) POST method (Receive) WSP Confirmed/Unconfirmed Push

4 Evaluation of WAP 1.x Binary Encoding

WAP 1.x defines two types of binary encoding functions: WML binary encoding and WSP header compact encoding. Since these functions are peculiar to WAP 1.x, we evaluated both of them.

4.1 Evaluation of WML Binary Encoding

The WAP Gateway encodes WML contents (WML tags as well as control codes) after XML parsing. Since the effectiveness of the binary encoding depends on the content, we evaluated the encoding rate by using several typical examples of WML contents as shown in Figure 3. The contents used essential tags such as “wml”, “card” and “p” and text data in addition to line break tags (i.e., “br”). We evaluated the binary encoding of WML contents using several content set sizes (i.e., 500, 1000, 1400, 20K, 100K, 360K bytes) by changing the size of the data part. The evaluation focused on the following two factors.

- Compression rate comparison between WML binary encoding and gzip
- Time taken for WML compression, which includes XML parsing time.

The result of the evaluation is shown in Figure 4. In the evaluation, we used RXP beta 15 as the XML parser.

```
<?xml version="1.0"?>
<!DOCTYPE   wml   PUBLIC   "-//WAPFORUM//DTD WML   1.1//EN"
   "http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
 <card id="card2" ontimer="./auto500-2.wml#card1">
 <timer value="50"/>
  <p mode="wrap">
   *Test of 1400 Octet Contents*<br/>
   ***deck1***<br/>
   WAP  is  a  protocol  that  is  designed  for  wireless
   environment.<br/>
   WAP  is  a  protocol  to  access  to  the  Internet  from
   cellular phones, PDAs and so on. <br/>
  (The rest is omitted.) </p>
 </card>
</wml>
```

Fig. 3. Example of WML content. It uses essential tags such as “wml”, “card” and “p” and text data in addition to line break tags (i.e., “br”).

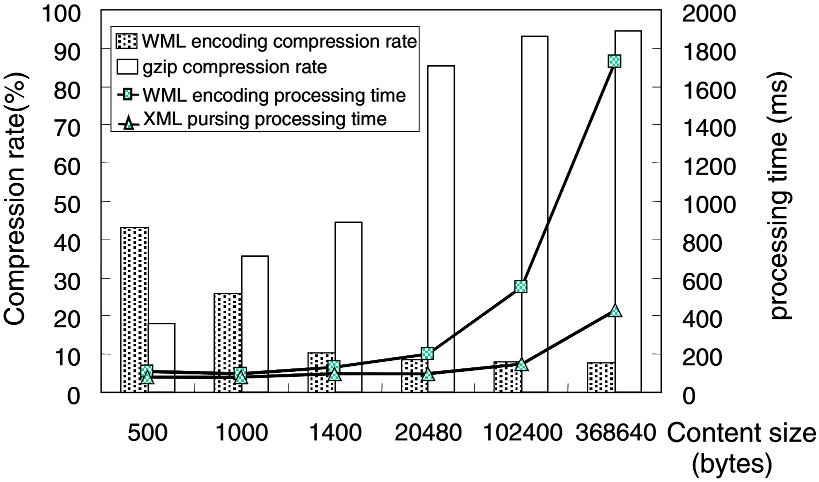


Fig. 4. The bars show WML encoding compression rate and gzip compression rate. The lines show WML encoding processing time and XML parsing time. This paper defines the compression rate as the ratio of the original WML content size to the reduced size of compressed content.

WML binary encoding is effective only if the content set size is small. The reason for this is that the XML and DOCTYPE declarations, which offer very high compression rates (i.e. 115 octets → 3 octets), are a significant fraction of the data only if the data set is small.

As for the WML encoding processing time, both WML encoding and XML parsing time increase rapidly with content set size (e.g., 1730 ms for 360 Kbytes). For a 500 bytes content set, 72.7% of the WML binary encoding processing time is occupied by XML parse time while the remainder is for the actual encoding. For a 368 Kbytes content set, however, the XML parse time occupies 24.9% of the WML binary encoding processing time. Therefore, actual binary encoding takes longer than XML parsing if the content set size is large.

WML binary encoding is not so effective if high-speed networks are used. For example, a 1400 bytes content set is binary encoded to 1117 bytes. The transmission delay of the size reduction (i.e., 283 bytes) is about 6 ms at 384 Kbps, and 236 ms at 9600 bps. Because it takes approximately 130 ms to encoding a 1400 bytes WML data set, encoding is only effective if the bearer network speed is low, i.e. 9600 bps.

We conclude that WML binary encoding offers some benefits but not for high-speed networks and/or large content sets. Thus it appears that WML binary encoding is not so effective since 3G networks offer high speeds and will be used to send large content sets such as multimedia data files.

4.2 WSP Header Compact Encoding

One unique function of WSP is WSP header compact encoding, which aims to reduce the size of the WSP header. Our test bed system took less than 10 ms to compress (185 bytes → 50 bytes) the WSP header on a connect message. Once again, compression is not effective if the network's speed is high. For example, if the network speed is 384 Kbps, the savings of 135 bytes is equivalent to about 3 ms. Obviously it is better to send the message without using WSP header encoding. A similar argument can be made for large data sets. If the network speed is 9600 bps then the transmission delay is approximately 112 ms, so WSP header encoding is effective. Given that 3G offers high speeds and will handle large data sets, WSP header encoding is problematic.

5 Comparing WAP 1.x to HTTP/TCP

Since WAP defines many specifications that vary from protocols to application environments, it is difficult to evaluate the overall performance of the WAP protocol. As the first step, we evaluated WAP functions (WAP 1.1) and their performance.

5.1 Functional Comparison between WAP 1.x and HTTP/TCP

While TCP is a byte-stream connection-oriented transport protocol, WTP provides transaction-type message transmission so that WAP 1.x can provide reliability in case of using WTP since WDP doesn't provide any reliability. While 3-way handshaking is necessary to establish a TCP connection, WSP establishes a session by using WSP connect and WSP connect reply messages. WTP can optionally send asynchronous transaction requests so that the WTP initiator can send subsequent packets without waiting to receive acknowledgements of the previous packets. This asynchronous transaction request function enables the WAP 1.x user to shorten the total communication time.

An example of the communication sequence of each protocol is shown in Figure 5. This example shows the case where the client receives one content set in one packet. Both protocols consist of:

- WSP session/TCP connection setup phase,
- Content pull phase and
- WSP session/TCP connection closure phase.

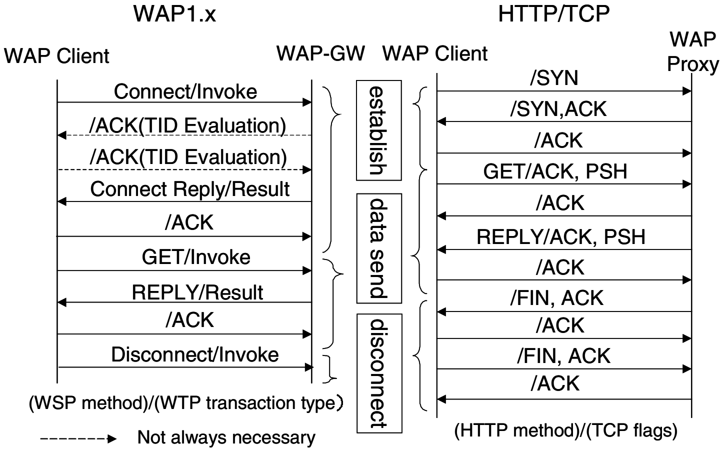


Fig. 5. Communication sequences of WAP 1.x and HTTP/TCP. The left and right figures are independent.

Since TCP offers the HTTP persistent connection defined in [7], setting up a new connection is not necessary if another connection is already established. WAP 1.x provides the same function by WSP.

5.2 Performance Evaluation Parameters

We choose three content set sizes (1K, 10K, 100K bytes). The 1 Kbytes set represents the small graphic elements common in web pages, and it is possible to send the set in one packet if the bearer Maximum Transfer Unit (MTU) is 1500 bytes. Since interactive users may see a page in segments and spend some time reading parts of it and clicking to see the next part of the page, it is the case for browsing such kind of small data. The 100 Kbytes set represents multimedia contents such as MPEG4 streaming data, and Java content. This kind of application is actually provided in Japan by the FOMA service. In case of WAP 1.x, large content sets that exceed one MTU can be conveyed if the SAR functionality of WTP is supported in WAP 1.x.

We used the parameters recommended in the WAP specifications of WTP 1.x in the test-bed system. The W-CDMA emulator parameters are the same as those described in section 3.

Of the TCP optimization techniques specified in [8], we used Selective ACK (SACK) [9], increased TCP initial window [10] and large receiver's advertised window. The last technique enlarges the maximum window size that can be estimated by the Bandwidth Delay Product (BDP) of the end-to-end path. The parameter of receiver's advertised window is one of the keys to improving transmission performance. For example, the IMT-2000 network can support up to 384 Kbps, and its Round Trip Time (RTT) of the end-to-end path varies from half a second to one second [8]. Therefore, the BDP of the end-to-end path over the IMT-2000 network can be very large. The sender cannot make full use of the network bandwidth if the receiver's advertised window size doesn't suit this BDP. In our early investigations, we found that the value of the receiver's window size should be set at appropriately 32 Kbytes (calculation based on BDP over FOMA) to maximize network performance. The RTT is estimated to be up to 667 ms if the window size is 32 Kbytes. We decide that this value is suitable considering the tradeoff between memory cost and expected performance gains. As for the WTP, we also used 32 Kbytes as the maximum group size parameter to be consistent with TCP.

5.3 Performance Comparison between WAP 1.1 and HTTP/TCP

We measured the response time as time between sending a request from the WAP client to receiving a response from the WAP gateway. We also measured the response time of HTTP/TCP. Note that neither of these response times includes connection setup time because we assume that the TCP persistent connection is used and a continuous WSP session is available.

The results of our measurement showed that the WAP 1.1 and HTTP/TCP have comparable performance (i.e. response time) except when transmitting larger contents (e.g. multimedia data types), in which case the performance of HTTP/TCP is better than that of WAP 1.x (Figure 6).

If we look at the result more in detail, we find that this is related to the difference in the flow control mechanisms of WTP and TCP.

- For 1 Kbytes set: HTTP/TCP is slightly better than WAP 1.1.

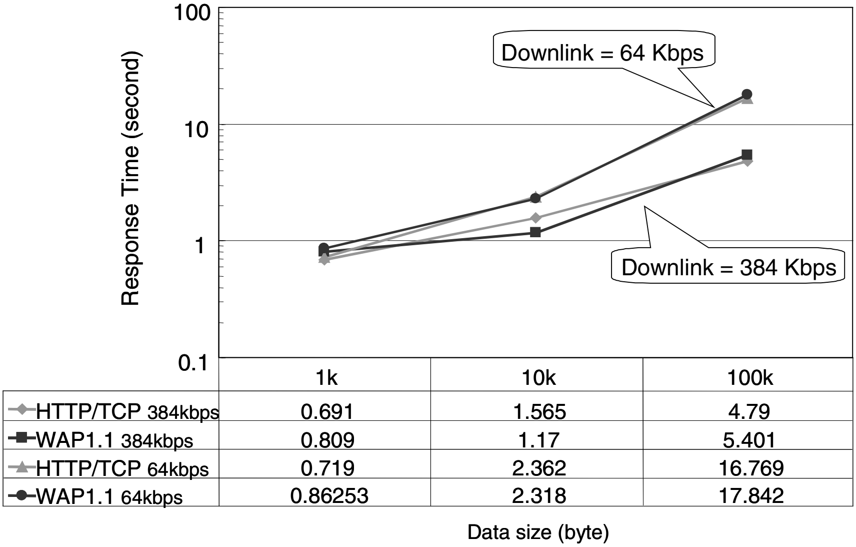
The reason of this is that the TCP slow start mechanism doesn't matter because the 1 Kbytes set can be sent in one packet. The difference between WAP 1.1 and HTTP/TCP is due to the overhead of WAP 1.1 protocol conversion at the WAP gateway (i.e., from WAP 1.1 protocol stack to Internet protocol stack). However, the impact on performance is insignificant.

- For 10 Kbytes set: WAP 1.1 is slightly better than HTTP/TCP

This is due to the difference in the flow control algorithms. In WTP, seven packets are used to send a 10 Kbytes set so the WAP gateway transmits these seven packets simultaneously within the maximum group size. However, the TCP slow start algorithm generally prevents full use being made of network resources, since TCP can send only two packets at the beginning of communication (assuming that the increased TCP initial window [10] is used). Therefore, the window size of TCP is not fully expanded when the sender finishes sending the 10 Kbytes set.

- For 100 Kbytes set: HTTP/TCP is better than WAP 1.1

If the packet size is as large as 100 Kbytes, TCP communication is finished after the TCP window size is fully expanded and adjusted appropriately based on underlying bearer's speed. In WAP 1.1, the sender has to wait until the sent packets are acknowledged. This incurs a slight waiting time because the WAP-GW needs to wait for a single ACK packet (Figure 7).



	1k	10k	100k
HTTP/TCP 384kbps	0.691	1.565	4.79
WAP1.1 384kbps	0.809	1.17	5.401
HTTP/TCP 64kbps	0.719	2.362	16.769
WAP1.1 64kbps	0.86253	2.318	17.842

Fig. 6. The response times of WAP 1.1 and HTTP/TCP. Note that both axes have logarithmic scale.

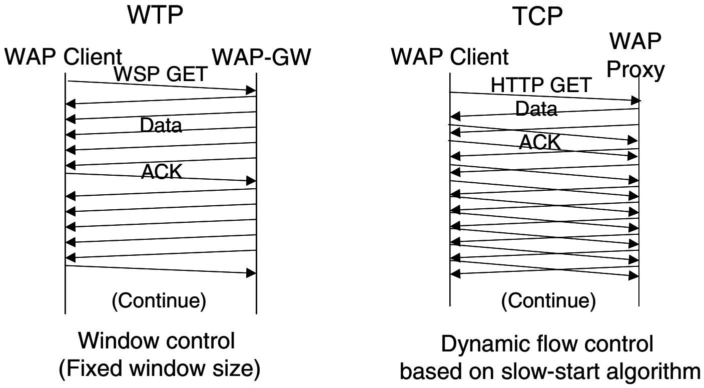


Fig. 7. In this example of WTP usage, there is waiting time after sending data packets until the ACK packet is received. The WAP-GW sends five packets continuously and waits for ACK packet reception since this example assumes that the receiver window size is 32 Kbytes (the number of packets is calculated by dividing the receiver window size by the MTU size). In this example of TCP usage, the ACK packet is piggybacked. The window size is not extended during at the beginning of communication because of the slow start mechanism.

From the results of our evaluation, we feel that TCP is applicable for high-speed wireless networks such as 3G since they are more likely to transmit large content sets such as multimedia data files.

6 Proposed Mobile Internet Architecture for 3G Wireless Networks

As mentioned in sections 4 and 5, WAP 1.x is appropriate for low-speed networks such as 1G and 2G since WAP 1.x has several encoding technologies to reduce the amount of data size and it also offers protocol optimisation. However, HTTP/TCP offers better performance over high-speed wireless networks such as 2.5G and 3G. Accordingly, our mobile Internet architecture, intended for high-speed wireless networks such as 2.5G and 3G (Figure 8), that adopts HTTP and Wireless Optimised TCP (W-TCP) [8], while TLS is used to achieve transport layer security. XHTML [11] is adopted as the markup language because it is expected to become the next generation standard markup language.

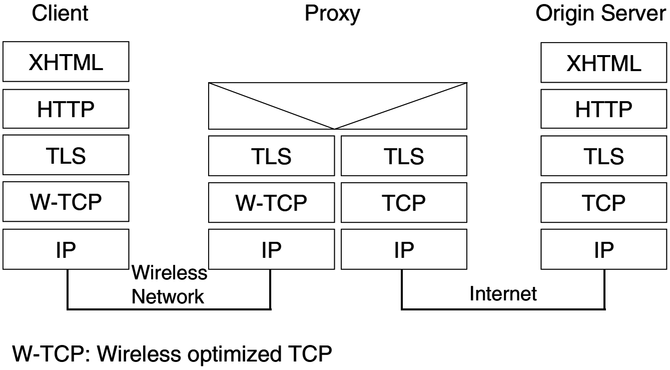


Fig. 8. Proposed mobile Internet architecture for 3G wireless networks. The proxy terminates the wireless-optimized TCP and acts as a proxy. TLS is used to achieve the end-to-end security needed to access the origin server directly through the proxy.

Since the WAP 1.x protocol is different from what is used in the Internet, it is theoretically unable to provide end-to-end security. If, however, we adopt HTTP/TCP, TLS can be used so the client can communicate with the origin server by using TLS tunneling through the WAP proxy.

One of the major reasons for the success of i-mode[2] is that it supports an HTML subset as the markup language so that it is compatible with the contents of the Internet. We think that it is easy to shift from HTML to XHTML because it is easy to create XHTML contents by anyone who has some skill in HTML content generation because XHTML is just a rewrite of HTML following the XML syntax. Moreover, XHTML is more flexible and extensible than HTML since it can be extended based on XML.

[2] For further information, please see (http://www.nttdocomo.co.jp/english/).

7 Conclusions

We investigated the performance of WAP 1.x by comparing it against the Internet protocol. Our result has shown that both protocols have comparable performance (i.e. response time) except when transmitting large content sets (e.g. multimedia data files), in which case HTTP/TCP offers better performance than WAP 1.x. We have also evaluated some WAP-specific functions such as the binary encoding of WAP headers and contents. Binary encoding is effective for small content sets, but its effectiveness is questionable for large content sets. Therefore, we concluded that the Internet standard protocol (i.e., HTTP/TCP) is suitable for high-speed wireless environments. We proposed a mobile Internet architecture for high-speed wireless networks such as 2.5G and 3G.

We have submitted our proposal to the WAP forum, which adopted it in August 2001 as the next version of WAP 1.x, called WAP version 2.0 (WAP 2.0) [1]. It was developed for high-speed wireless networks such as 2.5G and 3G, and its goal is to achieve Internet convergence.

Future research should look at fourth-generation (4G) mobile Internet services and protocols. Since 4G networks can support up to 100 Mbps, the services provided over such high-speed networks must be changed drastically. Considering the applicability of WAP 2.0 or other Internet-based protocols to the 4G wireless networks will be a future task and we plan to investigate the future mobile Internet architecture and applications.

References

[1] WAP Forum Specifications, http://www.wapforu.org/.

[2] P. Chaudhury, et al., "The 3GPP Proposal for IMT-2000," IEEE Communications Magazine, December 1999.

[3] M. Metter, et al., "WAP enabling existing HTML applications," Proceedings First Australasian User Interface Conference, January 2000.

[4] S. Gordon, et al., "Analyzing the WAP Class 2 Wireless Transaction Protocol Using Coloured Petri Nets," Proceedings of the 8th International Aerospace Congress incorporating the 12th National Space Engineering Symposium, September 1999.

[5] Thomas Kunz, et al., "WAP traffic: Description and comparison to WWW traffic," Proceedings of the Third ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM 2000), August 2000.

[6] 3rd Generation Partnership Project (3GPP) "RLC protocol specification," September 2001.

[7] R. Fielding, et al., "Hypertext Transfer Protocol - HTTP/1.1," RFC 2616, June 1999.

[8] H. Inamura, et al., "TCP over 2.5G and 3G Wireless networks," Internet-draft, February 2002.

[9] S. Floyd, et al., "An Extension to the Selective Acknowledgement (SACK) Option for TCP," RFC 2883, July 2000.

[10] M. Allman, et al., "Increased TCP's Initial Window," RFC 2414, September 1998.

[11] World Wide Web Consortium (W3C), "XHTML 1.0: The Extensible HyperText Markup Language," W3C Recommendation, January 2000.

Performance Evaluation of H.263–Based Video Transmission in an Experimental Ad–Hoc Wireless LAN System[*]

Matías Freytes

Laboratorio de Comunicaciones Digitales
Universidad Nacional de Córdoba
Argentina
mfreytes@gtwing.efn.uncor.edu

Abstract. This paper analyzes different packetization strategies that significantly improve the quality of H.263 coded video transmission in wireless local area networks (WLANs). We show that a considerable improvement can be obtained with the proper combination of error concealment techniques and transmission unit (TU) sizes. Moreover, we present performance evaluation results on critical system parameters for interactive video over Ad–Hoc WLANs, and propose a simple rule to specify TU sizes. We use *Kinesis*, a novel system architecture for packet video, as a software measurement tool to analyze the effects of packetization policies, distance, network offered load, and interference from co–located WLAN devices on overall video quality. *Kinesis* supports IP multicast extensions, overcoming delay issues introduced by the complex retransmission schemes in the IEEE 802.11 MAC sublayer, which are not acceptable for real–time services. It implements real–time transport protocol functions to manage synchronization and QoS, and performs software–only real–time H.263 video encoding.

In this paper we address most common Ad–Hoc WLAN configurations, and present experimental results on Packet Error Rates, Frame Error Rates, frame delays and latency, and Peak Signal–to–Noise Ratio for well–known test video sequences.

1 Introduction

The increasing trend toward networked portable computers, and recent advances in WLAN technology and video compression algorithms, have stimulated the demand for real–time video transmission services over wireless packet switched networks. Real–time interactive video systems over Ad–Hoc WLANs constitute a different and complex scenario that requires further analysis and experimental research. Physical layers supported by the IEEE 802.11 standard for WLANs have limited connection ranges, present higher error rates and have time–varying and

[*] This work has been supported in part by SeCyT, Universidad Nacional de Córdoba, and by a grant from the Agencia Córdoba Ciencia.

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 539–550, 2002.

asymmetric propagation properties. These characteristics have direct impact on performance degradation in real–time services: transmission systems in WLANs have to deal not only with network congestion as their wired counterparts, but also with corrupted packets due to higher bit error rates. To address these limitations, a complex Medium Access Control (MAC) protocol is required for adequate transport layer performance. The MAC protocols specified in IEEE 802.11 hide packet losses caused by bit errors by including a retransmission algorithm for corrupted packets which introduces significant delay and packet overhead [1].

Kinesis, a novel system architecture for packet video developed at our laboratory, overcomes this mechanism at the expense of higher frame dropping by forcing multicast addresses even in point–to–point communications. ARQ mechanisms introduce intolerable delays in interactive video systems. Although frame dropping results in video quality degradations, mechanisms are available to compensate for this effect, whereas nothing can be done about the excessive delays introduced by the IEEE 802.11 MAC ARQ. Link layer error checking (checksums) is useful for network services where no errors are tolerated, but it is too strict for applications where some degree of quality degradation is acceptable. Video applications should tolerate frame bit errors and be able to efficiently utilize non–corrupted regions of the bit stream, thus reducing information loss and enhancing overall video quality.

Recent publications investigate the performance of the transfer protocols proposed in this standard as well as voice and video services on packet switched infrastructure networks. In [2], Kamerman and Aben present throughput performance of 802.11 WLANs in relation to overhead from header fields in physical, medium access, and transport protocols. The authors focus on TCP file transfers in infrastructure WLANs, but do not address interference from co–located wireless networks, or real–time protocols and services. Weinmiller et al. ([3]) present comparative analysis on the performance of the access protocols in IEEE 802.11 and ETSI Hiperlan, regarding the impact of hidden stations, number of stations, and packet sizes. Quality of Service capabilities offered by point coordination function (PCF) access in 802.11 infrastructure networks are studied, but no results are presented for time–bounded services, or Ad–Hoc scenarios. Different performance–limitation factors in the two standards are considered in [4]. Two separate simulation scenarios are used, consisting of 10 and 100 stations organized in Ad-Hoc networks. Performance of MAC protocols is compared considering bit error rate (BER) of the fading air medium and the number of stations. However, no real–time experimental results are presented. Experimental throughput measurements comparing frequency hopping (FHSS) and direct sequence spread spectrum (DSSS) physical layers in several 802.11 infrastructure office environments are provided in [5]. Although the authors evaluate the effects of interference from adjacent WLANs, they do not address Ad–Hoc networks or time–bounded services. A high performance TCP protocol for lossy wireless links is presented in [6]. Congestion and random loss are differentiated, and window sizes, as well as TCP timers, are managed according to these two cases. Although this optimized protocol reports higher throughput and lower end–to–end delay, retransmission schemes in TCP are still too complex to be considered in time–bounded situations. Therefore, this is not a suitable solution for real–time interactive video applications. Bahl describes the challenges of supporting digital video in wireless networks in [7]. A custom video coding algorithm, a resource reservation scheme, and a software architecture are presented. The author focuses on the problem of providing high quality video on centralized wireless networks, but does not address Ad-Hoc configurations. Sachs et al. ([8]) propose an interesting hybrid ARQ system for streaming media over WLANs, and better performance results are reported for media servers attached through APs. However, this proposal is not suitable for interactive transmission due to buffering and decoding delays, and Ad-Hoc configurations are not discussed.

In this paper we analyze a packetization scheme that improves video quality at the receiver, and propose a simple rule to specify TU sizes. We also present performance evaluation results on critical system parameters obtained during *Kinesis* video sessions. We show the impact of distance between stations, interference from co–located WLANs, network offered load, and frame sizes on throughput and video quality. We measure Peak Signal–to–Noise Ratio (PSNR) and investigate the influence of INTRA and INTER frame errors by means of objective analysis.

The rest of this paper is organized as follows. In Section 2 we make a brief introduction to *Kinesis* and the general communication framework for networked multimedia. Section 3 discusses performance evaluation results, and in Section 4 we present our concluding remarks.

2 Kinesis

Kinesis is the real–time video transmission system developed at the Digital Communications Research Laboratory (DCRL). It was originally conceived as a software measurement tool to study the behavior of interactive video systems over WLANs. Initial successful results led to the development of a highly modular and extensible architecture, which would support RTP–based networked multimedia applications.

Kinesis is an object–oriented multi–threaded real–time video system made of autonomous and reusable modules. It incorporates our own implementation of the RTP protocols and our software–only H.263 video codec. It is easily extendable to accommodate new media types and offers an abstract interface to network connections. *Kinesis* supports IP multicast extensions, real–time video rendering on X terminals, video recording, and diverse video producers (frame grabbers, USB cameras, video disk files). Object oriented design and software construction are powerful tools to manage the complexity inherent in the development process of multimedia and networking distributed systems. Protocol modules, network connections and media codecs, for instance, are conveniently represented by system classes, providing a set of autonomous and reusable building blocks. Multi–threading led to the use of simplified abstract processors, in the form of active objects ([9] and [10]). It allows low latency on single–processor platforms, and efficient utilization of hardware resources on symmetric multi–processing computers. We have defined three such module–classes in a *Kinesis* session: media producer, media encoder, and media decoder, synchronized by media buffers, being the media encoder one of the most CPU–intensive tasks in a real–time video system.

2.1 Real–Time H.263 Video Encoder

H.263+ is the first international video coding standard specifically designed to work on different network technologies. It is a backward compatible extension of H.263 providing twelve optional modes to improve video quality in error-prone and non guaranteed QoS networks. Wenger et al [11] recommended and theoretically justified a combination of error-resilience optional modes for five scenarios based on wired networks. Although H.263+'s error-resilience modes were designed for wired networks they can be beneficial for wireless networks too, where video quality is severely degraded due to higher packet error rates.

Kinesis incorporates a new software–only H.263 encoder [12], which implements one of the most advanced discrete cosine transforms for fast video coding in interactive systems. Motion estimation is based on the Advanced Center Biased Three Step Search algorithm proposed in [13]. Compression tests on several well known video sequences report outstanding coding times. In Table 1 we present comparative results with the University of British Columbia's (UBC) H.263 Reference codec, version 3.2.

Table 1. Average video coding times (ms)

Frame type	Miss America		Susie		M & D		Foreman	
	DCRL	UBC	DCRL	UBC	DCRL	UBC	DCRL	UBC
INTRA	21.40	52.60	21.55	52.80	21.86	52.73	22.35	52.77
INTER	26.84	104.49	27.41	113.35	26.35	118.20	28.45	126.99

The video encoder incorporated in *Kinesis* performs much faster coding at the expense of a negligible impact on video quality, thus becoming an appropriate tool for interactive video conferencing systems. Although the current version only implements an H.263 video encoder, new media types can be easily added. Its flexible and modular architecture provides an excellent test–bed for new protocol proposals and makes it easily adaptable to new environments. In the next section we analyze performance results obtained during *Kinesis* video sessions.

3 Performance Evaluation

In this section we present experimental results obtained with *Kinesis* during video conferencing sessions over Ad–Hoc WLANs at the engineering building in our campus. We analyze the effects of video error concealment, data transmission unit sizes, offered network load, packet error rates (PER) and frame error rates (FER), INTRA and INTER frame losses, distance between stations, and interference from other WLAN devices in the same area, on the average video quality in wireless systems. We show how these system parameters affect the PSNR (1), as a measure of video quality. Although the average PSNR is an objective measure and may not reflect human perceived quality, it has been widely adopted as a distortion measure. In order to provide "subjective" information, we have also included some representative video frames for the case studies. We also include a detailed description of the equipment used, its set up, and measurement procedures.

In error–free transmissions, the PSNR of the video sequence reproduced at the receiver is given by:

$$PSNR = 10\log_{10}\frac{255^2}{\frac{1}{N}\sum_{i=1}^{N} D_{sc}(i)} \ , \tag{1}$$

where N is the number of frames in the sequence, and D_{sc} is the video source coding distortion given by:

$$D_{sc} = \frac{1}{n \times m}\sum_{x=1}^{n}\sum_{y=1}^{m}\left|C_i(x,y) - \hat{C}_i(x,y)\right|^2 \ .$$

$C_i(x,y)$ and $\hat{C}_i(x,y)$ are the transmitted and received frames, (x,y) the pixel coordinates in the frame, and $(n \times m)$ the frame size. Transmission errors introduce additional distortion at the receiver, known as channel distortion (D_{ch}). Thus, the overall distortion of the decoded video sequence is given by $D_{sc}+D_{ch}$:

$$PSNR_d = 10\log_{10}\frac{255^2}{\frac{1}{N}\sum_{i=1}^{N} D_{sc}(i) + D_{ch}(i)} \ , \tag{2}$$

The loss of picture quality, defined as:

$$\begin{aligned} \mathit{\Delta} PSNR &= PSNR_d - PSNR \\ &= 10\log_{10}\frac{D_{sc}}{D_{sc}+D_{ch}} \ , \end{aligned} \tag{3}$$

is used as a measure of video degradation.

3.1 Experimental Environment

The experimental Ad-Hoc WLAN configured at the engineering building is conformed by Pentium III desktop and laptop PC computers running GNU/Linux. The stations are equipped with DSSS IEEE 802.11b wireless medium interfaces configured on channel 3 (2.422 GHz). Video producers consist on file producers of the well–known video test sequences "Mother&Dougther", "Carphone", "Foreman", and "Deadline". Each one of the results represent approximately

50000 QCIF frames (176 × 144 pixels) at 15 f/s. Measurements were taken on the second floor of the building, which consists primarily of office and laboratory rooms.

In wireless Ad-Hoc scenarios, transmission errors degrade video quality at the receiver. INTRA coded video sequences provide good streaming quality at the expense of inefficient bandwidth usage. There is no error propagation, and encoding algorithms are simpler and faster. Differential (INTER) coding, on the other hand, allows better use of the available bandwidth, but suffers from significant performance degradation. In order to mitigate the effects of transmission errors, a periodic INTRA refresh has been widely used. ITU–T H.263 specifies INTRA Macro Block (MB) coding once every other 132 times the MB is encoded, and several authors propose INTRA frame updates (e.g. [14],[15]), also known as Full Intra Refreshments (FIR).

H.263 provides means to insert synchronization words at the picture level, and, optionally, at the GOB level. The latter allows resynchronization in case of errors and inserts GOB sync headers at the beginning of each MB row. This is exploited by the error concealment technique used in the decoder, which discards corrupted GOBs and replaces the corresponding image content with data from the previously decoded frame. Although this technique has demonstrated very good results for non–moving parts of the sequence, it introduces noticeable distortion in moving image regions.

In Fig. 1 we show a typical ΔPSNR profile obtained during *Kinesis* video sessions. We plot ΔPSNR and packet errors for 1370 "Deadline" frames. As we can see, picture distortion caused by transmission errors (i.e. channel distortion introduced by dropped RTP packets) is usually more noticeable than the distortion produced by the propagation of errors inherent in differential coding techniques. This characteristic has severe consequences in packet switched

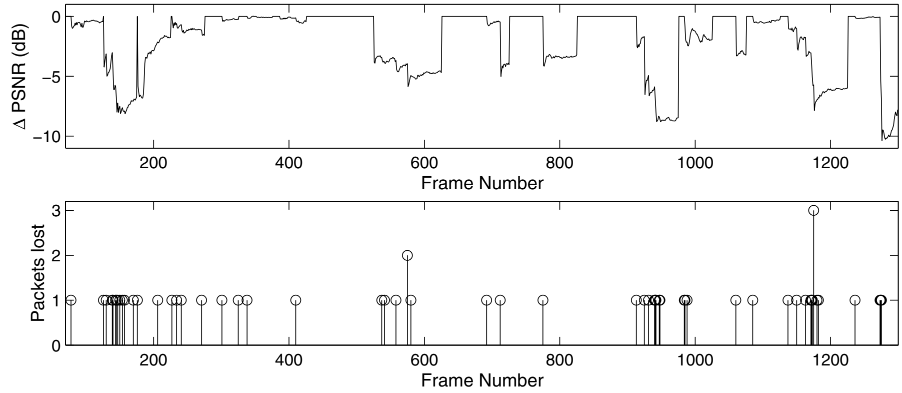


Fig. 1. ΔPSNR and packet errors per frame. Average ΔPSNR=-2.15 dB, FIR=50.

networks, where complete packets are generally lost by congestion on the network elements (e.g. routers), or at the local interfaces. Packet loss rates and their effects on overall video quality become critical in packet erasure channels with high bit error rates like the 802.11 Ad–Hoc WLAN under consideration. The consequences of transmission errors become more serious when corrupted data packets correspond to INTRA coded frames, producing deep PSNR peaks and seriously degrading quality. In Fig. 2 we show immediate error propagation effects after a lost frame compared to an error–free received sequence.

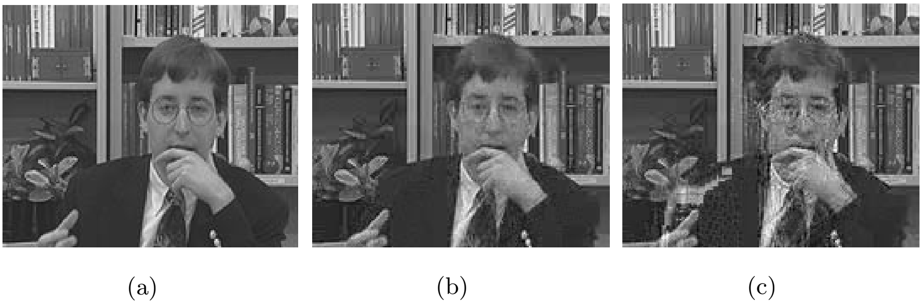

(a) (b) (c)

Fig. 2. Original frame 2(a), source–coding distortion 2(b), and source–coding plus channel distortion 2(c).

3.2 Experimental Results

The first set of experiments analyzes the effects of distance between Ad–Hoc 802.11 stations and interference from co–located WLAN systems on the average video quality for indoor applications. Towards this end we performed several measurements placing portable stations at 22 m, 44 m, 66 m, 88 m, and 110 m from each other. In all cases we verified average PSNR degradation less than 0.2 dB. The FER measured in INTRA coded sequences is an order of magnitude larger than the values obtained for INTER sequences due to larger frame sizes and longer bursts. However, less degradation is perceived because there is no error propagation. These results show that the effects of distance between stations in indoor configurations with line of sight is negligible.

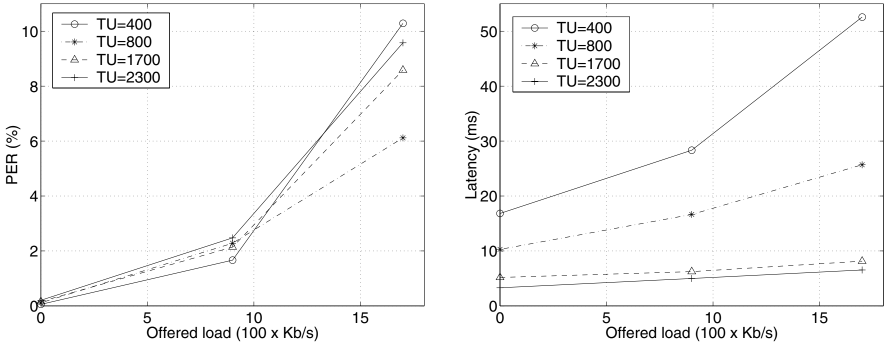
Then, we analyzed the influence of video transmission systems running over co–located 802.11 WLANs on FER and overall video quality. The interfering system was configured on channels 1, 5, 6, and 7, and the video session stations on channel 3 (2.422 GHz). In Table 2 we present average ΔPSNR and FER results for the tested channels. It should be noted that video quality is severely degraded when adjacent 802.11 channels are being used. These results demonstrate that a detailed planning scheme of channel reuse policies must be considered in order to provide acceptable digital video quality on Ad-Hoc 802.11 WLANs.

Table 2. Interference from adjacent WLANs

Channel	ΔPSNR (dB)	FER (%)
1 (2.412 GHz)	-12.84	53.35
5 (2.432 GHz)	-13.47	56.00
6 (2.437 GHz)	-0.41	0.6
7 (2.442 GHz)	-0.40	0.5

The second set of experiments analyzes the overall video quality at the receiver for different network loads TU sizes. We use INTRA and INTER coded sequences with FIRs at 44 and 132 frames. In both cases, we use a GOB replacement concealment mechanism to mitigate the effects of transmission errors. Our conclusions propose a simple rule to set the TU size for INTRA and INTER–coded real–time video in 802.11 WLAN systems.

INTRA–coded sequences. In Fig. 3(a) we show the effects of network load on the PER for different TU sizes. Our results show how small–TU error rates rapidly rise, even above medium and large TU values, as network load increases. At high offered loads, medium size TUs outperform small and large TUs. This is justified by the fact that INTRA–coded sequences, with large average GOB sizes (i.e. 300 Bytes) demand many small TUs to transport one video frame. Arrival time for all of the packets conforming one image is time bounded to guarantee acceptable decoding and rendering times at the receiver, thus forcing packet bursts at the transmitter (see Fig. 3(b)). Large TUs, on the other hand, require fewer packets to convey a full video frame. However, we also show that large TUs are more vulnerable, under low network traffic, than small ones due to wireless channel impairments.

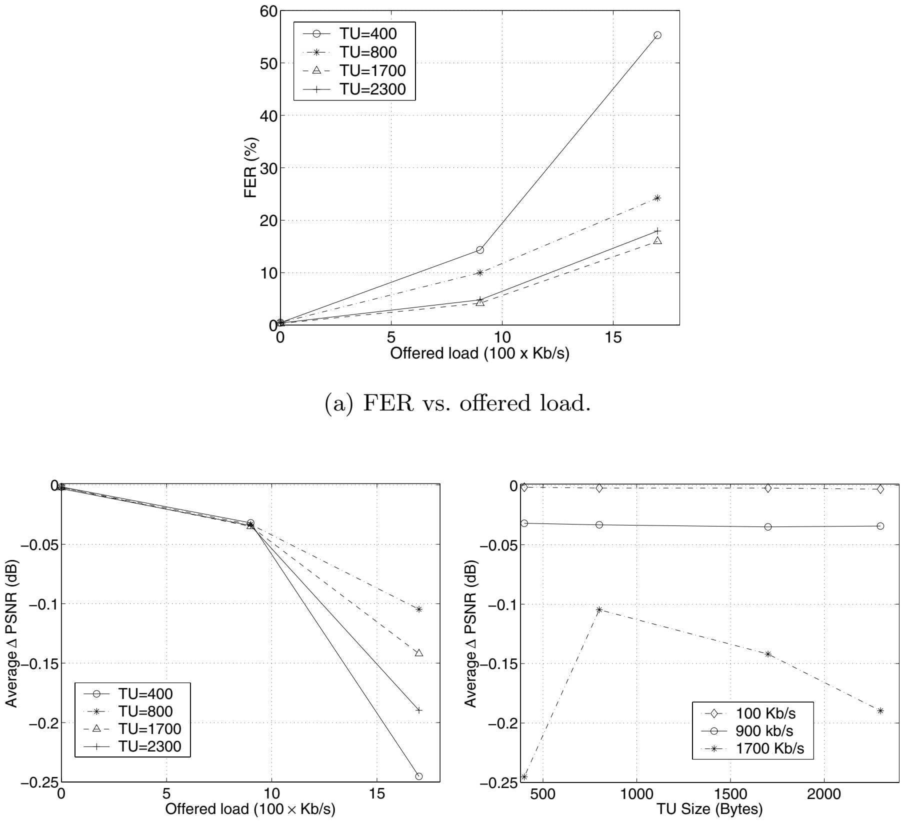


(a) PER vs. offered load.

(b) Frame latency vs. offered load.

Fig. 3. PER and latency vs. offered load for INTRA–coded sequences.

Fig. 4 analyzes the overall ΔPSNR for the previous case studies. We verified that medium and large TUs outperform small ones by an almost negligible gain as offered load increases. Although high FERs have been reported under heavy network load conditions for small TU sizes (Fig. 4(a)), the ΔPSNR is still acceptable. INTRA–coded sequences guarantee no error propagation, and the error concealment technique previously described reduces the impact of lost packets, improving overall video quality. If we use frame level synchronization with a follow–on encapsulating packetization policy, a single bit error in the transmitted data will damage the whole frame since there are no resynchronization points. No error concealment methods are usually used, and the FER gives a complete idea of the received video quality. However, when using GOB–



(a) FER vs. offered load.

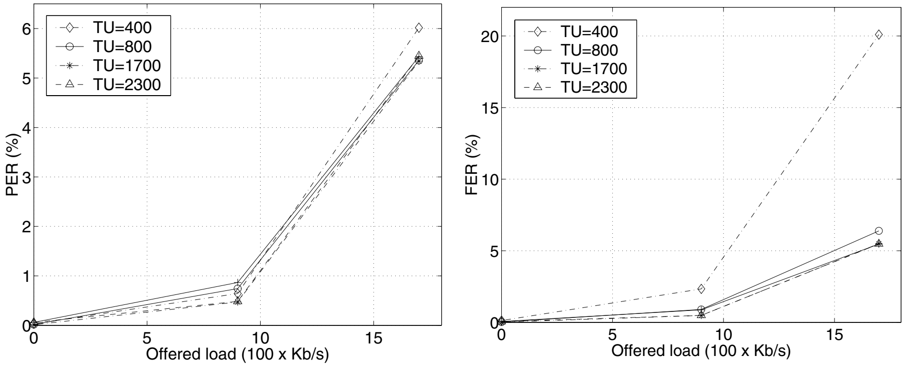
(b) ΔPSNR vs. offered load.

(c) ΔPSNR vs. TU size.

Fig. 4. ΔPSNR for INTRA–coded sequences.

level synchronization, although the FER could become larger, measured PSNR could indicate better performance results. Unlike the previous case, a bit error in the transmitted frame just damages the GOB containing the corrupted bit, and the receiver can resync at the next GOB. This way, only damaged GOBs have to be replaced with the corresponding GOBs from previous frames, keeping non–damaged GOBs in the current frame.

INTER–coded sequences. The following results correspond to INTER–coded sequences with FIRs at 44 and 132 frames, and sync at the GOB level. Although we allow large TUs, the whole unit would only be used in FIRs. INTER–coded frames are significantly smaller (average GOB sizes from 40 to 50 Bytes) and the TU adapts itself to these smaller sizes. For this reason, we introduce a smaller TU (100 Bytes) and the results are shown in Fig. 5. It should be noted that all of the TUs present similar performance results, except for TU=100, where packet transmission bursts can not be avoided.



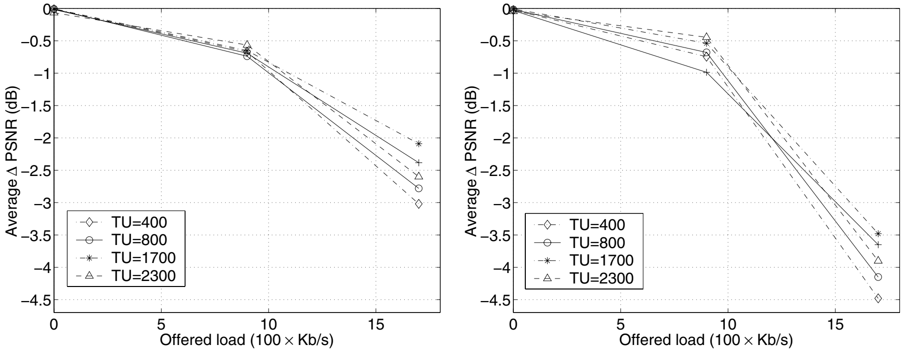
(a) PER vs. offered load.

(b) FER vs. offered load.

Fig. 5. PER and FER vs. offered load for INTER–coded sequences (FIR=132).

In Fig. 6 we present the average ΔPSNR for INTER–coded video sequences. According to the results presented in Fig. 5(b), we verified that the overall video quality for medium and large TUs outperforms quality obtained with small TUs. Note that this difference becomes smaller as the offered load of the system decreases, and access to the transmission medium becomes more reliable.

When the system is in an unloaded condition, the channel distortion (D_{ch}) is negligible for all TU sizes since the FER is very small. However, ΔPSNR values for both, FIR=44 and FIR=132, report considerable degradation in overall video quality introduced by channel distortion as network offered load rises.



(a) FIR=44.

(b) FIR=132.

Fig. 6. *Δ*PSNR vs. offered load for INTER–coded sequences.

At high load levels, an average advantage of 1 dB can be obtained for most common test sequences when selecting correct TU sizes. This video quality improvement becomes larger for complex sequences such as “Carphone”, where we measured a 2 dB gain. TUs should be large enough to avoid packet bursts, and sufficiently small to overcome wireless channel impairments. The TU should be set to a multiple of the sync block size (i.e. GOB), avoiding frame encapsulation, although this incurs in longer frame latencies at the receiver.

4 Summary

In this paper we presented experimental results for real–time video transmission in Ad–Hoc WLAN systems. We have considered packetization schemes for several network load conditions and video coding policies, impact of distance between wireless stations, and interference from adjacent WLAN systems.

We have analyzed the effects of these system parameters on overall video quality and presented objective results based on *Δ*PSNR, FER, and frame latency. We have shown that a significant improvement can be achieved with proper error concealment techniques and packetization strategies, and we proposed a simple rule to set TU sizes. We have also discussed the effects of INTRA and INTER frame errors by means of objective and subjective analysis.

Frame error rates and their effects on overall video quality become critical in packet erasure channels with high bit error rates like the IEEE 802.11 WLANs. Current 802.11 MAC sublayer protocols implement an error checking mechanism (FCS) which completely discards whole frames when bit errors are detected. Interactive video applications over WLANs can take advantage of error correction and error resilience techniques like RESCU [16] to protect INTRA–coded frames or to alleviate error propagation. Networked video applications can partially compensate for higher bit error rates in wireless environments by capturing corrupted payload data and either correcting bit errors by means of a FEC scheme, or identifying non damaged regions. This implies an integral solution affecting link, transport and application layer protocols, together with new video coding techniques. We are currently working on 802.11 MAC error–checking policies and on lightweight real–time transport protocols to provide the necessary support for interactive video over WLAN networks.

References

1. IEEE Study Group 802.11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. (1997) Draft Standard P802.11D5.3.
2. Kamerman, A., Aben, G.: Net throughput with ieee 802.11 wireless lans. In: IEEE Wireless Communications and Networking Conference (WCNC). (2000) 747–751
3. Weinmiller, J., Schlager, M., Festag, A., Wolisz, A.: Performance study of access control in wireless lans – ieee 802.11 dfwmac and etsi res 10 hiperlan. Mobile Networks and Applications **2** (1997) 55–67
4. Hadzi-Velkov, Z., Spasenovski, B.: Performance comparison of ieee 802.11 and etsi hiperlan type 1 under influence of burst noise channel. In: IEEE Wireless Communications and Networking Conference (WCNC). (2000) 1415–1420
5. Henty, B.E., Siew, J., Rappaport, T.S.: Frequency hop spread sperctrum and direct sequence spread spectrum study in ieee 802.11 and 802.11b wireless lan systems. Mobile and Portable Research Group – Virginia Tech (2001)
6. Parsa, C., Garcia-Luna-Aceves, J.J.: Differentiating congestion vs. random loss: A method for improving tcp performance over wireless links. In: IEEE Wireless Communications and Networking Conference (WCNC). (2000) 90–93
7. Bahl, P.: Supporting digital video in a managed wireless network. IEEE Communications Mag. (1998) 94–102
8. Sachs, D.G., Kozintsev, I., Yeung, M., Jones, D.L.: Hybrid arq for robust video streaming over wireless lans. In: Information Technology: Coding and Computing (ITCC). (2001) 317–321
9. Meyer, B.: Object-oriented Software Construction. Second edn. Prentice Hall International (1997)
10. Gibbs, S.: Composite multimedia and active objects. In: ACM SIGPLAN Workshop on Object-Based Concurrent Programming (OOPSLA). (1991) 97–112
11. Wenger, S., Knorr, G., Ott, J., Kossentini, F.: Error resilience support in h.263+. IEEE Transactions on Circuits and Systems for Video Technology **8** (1998) 867–877
12. Eschoyez, M., Freytes, M., Rodríguez, C.: Hardware–independent h.263 video encoder for interactive video conferencing systems. In: Multimedia Technology and Applications Conference (MTAC). (2001) 89–94
13. Nisar, H., Choi, T.S.: An advanced center biased three step search algorithm for motion estimation. In: IEEE International Conference on Multimedia and Expo. Volume 1. (2000) 95–98
14. Turletti, T., Huitema, C.: Videoconferencing on the internet. IEEE Trans. on Networking **4** (1996) 340–350
15. Chan, N., Mathiopoulos, P.: Efficient video transmission over correlated nakagami fading channels for is–95 cdma systems. IEEE JSAC **18** (2000) 996–1010
16. Rhee, I., Joshi, S.R.: Error recovery for interactive video transmission over the internet. IEEE JSAC **18** (2000) 1033–1049

Differentiated Services Based Priority Dropping and Its Application to Layered Video Streams

Markus Fidler

Chair of Computer Science IV, RWTH Aachen,
Ahornstr. 55, 52074 Aachen, Germany
fidler@i4.informatik.rwth-aachen.de

Abstract. In this paper we report on an implementation and evaluation of Internet priority dropping schemes and their application to layered video transmissions. The incremental decoding of each of the different layers of such an hierarchical video stream leads to an enhancement of the video quality, whereas the absence of a layer renders the receipt of higher layers useless. Thereby the layers have a specific order of precedence, which reflects their importance on the video quality and on the decoding process. We accommodate this hierarchy by mapping the video layers on different traffic classes implemented in a Differentiated Services network. We present a thorough evaluation of these schemes and we demonstrate the performance gain, if different Quality of Service classes for the transmission of the different layers are applied. Further on we address the interaction between non-responsive video flows and responsive streams and show how fairness can be supported by this approach.

1 Introduction

The convergence of telecommunication systems and data networks requires services beyond what is currently provided as Best-Effort (BE) service by the Internet. Especially real-time audio and video streams have significantly different traffic properties and requirements than common web or file transfer traffic. For example for high quality video-conferencing Quality of Service (QoS) parameters like throughput, delay, and drop rate are critical for the performance. On the other hand these applications commonly use the User Datagram Protocol (UDP) and the Real-time Transmission Protocol (RTP) [24]. They are often not responsive to congestion and tend to steal capacity from responsive Transmission Control Protocol (TCP) flows. Thus also application adaptivity and TCP-friendly rate control are required.

Since the first Internet congestion collapse a number of mechanisms that address congestion have been developed. These are mainly TCP slow start and congestion avoidance [1], which are supported in the network by Random Early Detection (RED) [7]. But also for non TCP flows, effort is made to standardize congestion control [8], both to enhance network performance and to increase fairness, if these flows compete with responsive TCP flows.

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 551–562, 2002.

Congestion is in the absence of Explicit Congestion Notification (ECN) [20] indicated by packet drops. If packets are lost, mainly three options exist for a real-time audio or video application: Dropped packets can simply be ignored, thus leading to a quality reduction, Forward Error Correction (FEC) [22] can be applied to recover from further lost packets, or Automatic Repeat Request (ARQ) techniques can be used for retransmissions [15,18]. FEC and ARQ mechanisms do increase the senders data rate. In order to compensate for this effect and to reduce congestion, a rate adaptation must be performed [27]. An option is to apply FEC or ARQ selectively to protect only the more important parts of a stream [18,25]. An hierarchical separation of video data into a set of layers [4,13] allows for the desired differentiation in terms of importance. Layered video also serves in this context for coarse congestion control schemes [21], and multi-cast scenarios with heterogeneous clients [16].

Apart from a protection by FEC or ARQ, the network can protect streams or sub-streams by applying priority dropping. A comparison of uniform and priority dropping schemes for layered video is given in [2]. Analytic and simulative results are shown for performance and incentive properties. The authors find that in case of the applied utility functions priority dropping does result in a performance gain, but they conclude that in the absence of proven utility models no definitive answer can be given. Nevertheless we have shown in simulations that in terms of the Signal-to-Noise Ratio (SNR) priority dropping performs better than uniform dropping [6]. Further on, we assume that instead of protecting parts of a stream by FEC or ARQ, it is better to apply different drop precedence level implemented by the network. Doing so also ensures the transmission of the more important parts of the stream, while less important parts can be dropped due to congestion. Moreover the disadvantages of FEC and ARQ like an increased data rate and an additional play out delay are not immanent to this approach.

In the future Internet an implementation of different drop rates can be based on the precedence field in the Internet Protocol (IP) Type of Service (ToS) field. A more advanced implementation of different drop precedence classes can apply Differentiated Services (DS) [3], which is the most recent approach of the Internet Engineering Task Force (IETF) towards QoS. The scalability issues encountered with the formerly proposed Integrated Services architecture are addressed by DS by a Class of Service (CoS) aggregation of individual flows to classes. The sender can mark packets to be of a certain priority in the RTP [19] header or, if the sender is aware of the QoS mechanisms provided by the network, it can perform the relevant marking, for example set the IP ToS field. It has to be noted that also in Asynchronous Transfer Mode (ATM) networks two drop priorities, which can be assigned to critical and non-critical data of a video stream, based on the ATM Cell Loss Priority (CLP) bit, exist [17].

In the remainder of this paper we focus on a real-time video application like video-conferencing with data rates in the range of a few tens of kb/s upto 5 Mb/s, and delay requirements of about 100 ms [14]. It is organized as follows: We give an overview on video coding in Sect. 2 and on DS in Sect. 3. In Sect. 4 we show our experimental setup and the obtained results and conclude in Sect. 5.

2 Video Coding

Common video coding standards like H.263+ [4] and MPEG-4 [13] implement video data compression by applying quite similar means. The data reduction is achieved by reducing redundancy and by dropping information, which is of less importance for the human visual system. This is performed concerning spatial information on single pictures (intra-frame), and concerning temporal information between different consecutive pictures (inter-frame). Temporal redundancy between consecutive frames is addressed by predicting pictures from earlier ones. Two types of predicted frames are used, forward predicted P-frames and bidirectionally predicted B-frames. In addition to the predicted frames intra-coded I-frames exist, which are encoded independently. Figure 1 shows a simplified video encoder consisting of Discrete Cosinus Transform (DCT), Quantizer (Q), and Entropy Encoder (EE) used for I-frames and P-frame difference pictures, and a similar decoder with Motion Estimation (ME) for the generation of the P-frames motion vectors.

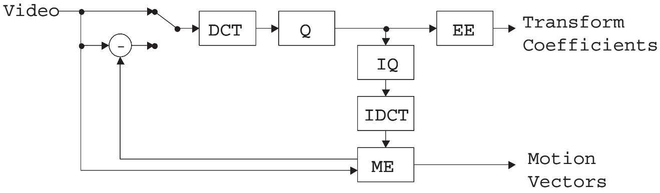


Fig. 1. Video Encoder

The quality of an encoded stream, the compression gain, and thus the data rate depend on the applied quantization parameter. The Peak (P) SNR according to 1 that is realized with the tmn H.263+ enocder is shown for different data rates in the top curve in Fig. 3 for the "apple" sequence. The encoder input is Quarter Common Intermediate Format (QCIF) with $i = 176 \times j = 144$ pixels with 8 bit depths and a luminance (Y) chrominance (U,V) sub-sampling of 4:1:1.

$$PSNR = \frac{i \cdot j \cdot Y_{max}^2}{\sum_i \sum_j \left(Y_{orig}(i,j) - Y_{dec}(i,j)\right)^2} \tag{1}$$

Current video coders allow for a hierarchical coding in a set of layers, which are classified into a base layer and a number of enhancement layers. The base layer contains all information that is necessary to display the video with a comparably low quality. On top of the base layer an incremental decoding of the enhancement layers is possible, whereas each successive layer improves the quality of the decoded stream. Three options of hierarchical encoding are defined in the H.263+ standard [4]. These are temporal, SNR, and spatial scalability as shown in Fig. 2. The arrows indicate the direction of prediction. In case of layered video it can be distinguished between forward prediction within one layer and upward prediction between the layers. In addition the MPEG standards allow for the option of data partitioning.

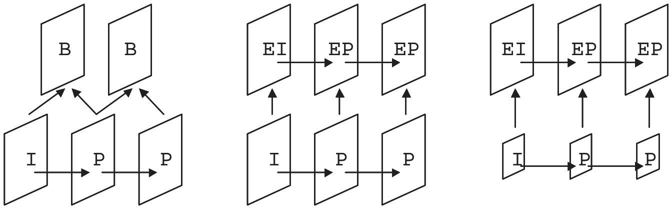


Fig. 2. Temporal, SNR, and Spatial Scalability

- *Temporal scalability* is achieved deploying the disposable nature of B-frames. Due to the high compression gain of B-frames the data rate of the enhancement layer is comparably low, but B-frames increase the play out delay.
- *Signal-to-Noise Ratio scalability* permits the refinement of the video in terms of the SNR. The quantization parameter of the encoder can be set independently for each layer and thus enables an encoding of the layers with independent target data rates, and an increasing SNR.
- *Spatial scalability* is achieved by encoding the frames with a low resolution for the base layer and with a higher resolution for the enhancement layers. Spatial scalability is apart from an up-sampling procedure implemented by the same means as SNR scalability.

An example of the PSNR of an encoded sequence applying SNR scalability is given in Fig. 4. In Fig. 5 and 6 a base and the corresponding enhancement layer picture are shown. Both sub-streams of this example are encoded with a rate of 30 frames per second and a data rate of about 50 kb/s.

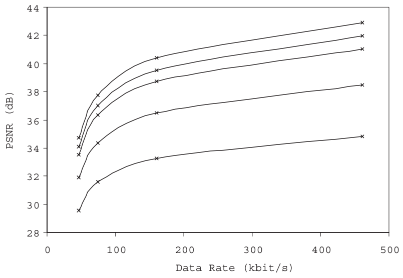


Fig. 3. Drop Rates {0;0.01;0.02;0.05;0.1}

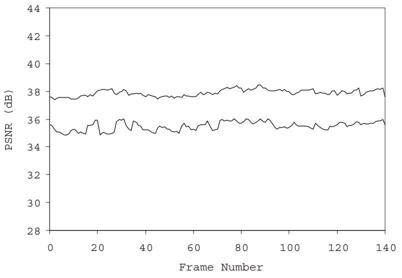


Fig. 4. Base/Enhancement Layer

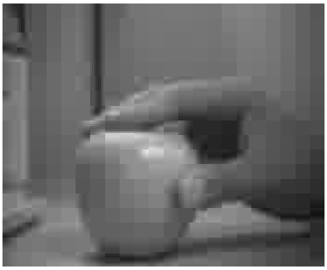


Fig. 5. Base Layer

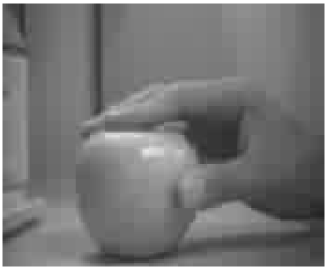


Fig. 6. Base+Enhancement Layer

Though video applications are considered to be loss tolerant [14], packet drops interfere with the decoding process and have adverse effects on the PSNR as shown in Fig. 3. Especially the effects of error propagation have to be considered. Due to the predictive encoding, errors propagate through the frame sequence up to the next synchronization point, namely the next I-frame. Thereby the robustness of the stream is directly influenced by the frame distance between consecutive I-frames n, whereas $n = 10$ in Fig. 3. To address these effects we describe frame loss and error propagation with a simple analytical model based on only two layers. A more general approach based on utility functions is given in [2], with the drawback that the generality limits the significance of the results. Equation 2 gives the probability of an error free decoding of a base layer frame i for a base layer frame loss probability p_b. The corresponding probability for enhancement layer frames is given in 3, with the respective loss probability p_e.

$$q_b(i) = (1 - p_b)^{((i \bmod n)+1)} \tag{2}$$

$$q_e(i) = ((1 - p_b) \cdot (1 - p_e))^{((i \bmod n)+1)} \tag{3}$$

With $p_b = p_e = p$, 2 and 3 describe the special case of uniform dropping. For reasons of simplicity we further on assume that the base and enhancement layer are encoded with a similar frame and data rate, which can for example be achieved by SNR scalability. Under the constraint $p = (p_b + p_e)/2$, which ensures that base layer frames can only be protected at the cost of a corresponding increase of the enhancement layer drop rate, a comparison of uniform and priority dropping can be investigated. We formulate the optimization problem in 4 for one Group of Pictures (GoP), with the weight w_b denoting the quality gain of an error free decoding of a base layer frame and w_e giving a similar weight for an enhancement layer frame. The decoding probabilities p_b and p_e are statistically dependent, such that the decoding of an enhancement layer frame implies the error free decoding of the relevant base layer frame. Therefore the resulting weight $w = (w_e - w_b)$ is applied for the enhancement layer. It can be assumed that $w_b > (w_e - w_b)$ in terms of PSNR, as can be seen from Fig. 3.

$$\max \left(\frac{w_b}{n} \sum_{i=0}^{n-1} q_b(i) + \frac{w_e - w_b}{n} \sum_{i=0}^{n-1} q_e(i) \right) \tag{4}$$

Since $p_b + p_e = 2p \ll 1$ we apply the approximation $(1 - p_b)(1 - p_e) \approx 1 - 2p$. Thereby the term that denotes the quality gain of a successful decoding of the enhancement layer does not depend on the individual drop probabilities of base and enhancement layer, whereas the term that denotes the quality gain of a successful decoding of the base layer depends only on p_b. The maximum according to 4 is then found for $p_b = 0$ and $p_e = 2p$, thus indicating the advantage of priority compared to uniform dropping.

3 Differentiated Services

A DS network [3] is composed of ingress respective egress nodes at the edges and core nodes within the DS domain. At ingress nodes a classification of the traffic into the different classes, which are supported within the domain, is performed. The respective class or behavior aggregate is marked as DS Code Point (DSCP), which supercedes the IP ToS field. Apart from this marking, the incoming traffic can be metered against some defined traffic characteristics usually by applying a token bucket, it can be policed or dropped, if certain traffic parameters are exceeded, and traffic shaping can be applied, to limit traffic bursts of certain classes that enter the DS domain, for example by applying a leaky bucket.

Within the DS domain each DSCP is mapped on a Per Hop Behavior (PHB). A share of the link capacities is reserved for each of these PHB. The PHB specify the service that is applied for the pertaining packets on their way to the next hop. So far one PHB and one PHB group have been defined. The single PHB is the so called premium service Expedited Forwarding (EF) [9], which provides low loss, low delay, and low delay jitter. A recommended implementation of EF is Priority Queuing (PQ). The PHB group is Assured Forwarding (AF) [9] and consists of four independent forwarding classes, whereas each class allows for a differentiation of upto three levels of drop precedence. Usually the AF classes are realized by a Weighted Fair Queuing (WFQ) environment, whereas Multiple (M)-RED implements the different levels of drop precedence.

AF packets can be marked by an ingress marker as green, yellow, or red depending on whether they conform to some specified traffic characteristics or not. A marking can be performed by a Single Rate Three Color Marker (SRTCM) [10]. The SRTCM is based on a token bucket with a Committed Information Rate (CIR), a Committed Burst Size (CBS) and an Excess Burst Size (EBS). A packet is marked green, if it does not exceed the CBS, yellow if it exceeds the CBS but not the EBS, and red otherwise. An alternative implementation is the use of a Two Rate Three Color Marker (TRTCM) [11], which marks a packet red, if it exceeds a Peak Information Rate (PIR), yellow, if it does not exceed the PIR but the CIR and green otherwise. Both of the TCM can be used either in the color-blind mode, in which an incoming stream is assumed to be uncolored, or in the color-aware mode, if packets are pre-colored for example by an upstream DS domain or by the sending host itself. Within the DS domain the color of the packet determines the mapping of the packet on the applied drop precedence.

Besides the EF and AF PHB a so-called Scavenger Service (SS) [26] was proposed in the DS environment recently. This service forms a less than BE service, which allows applications to utilize unused capacity, while the BE traffic is protected by assigning only a small weight of the link capacity to the SS. The SS thereby can allow for the transmission of high data rate and even unresponsive flows ideally without interfering with the BE service. A typical SS implementation is WFQ with a minimum link share that is configured to 1 %.

Figure 7 and 8 show the DS configuration of a core router. The EF, one of the four possible AF, the BE, and the SS queues are shown in the model. Besides two possible M-RED configurations are given, one with only two drop precedence level for a differentiation of green and red packets within an AF class, and one that in the absence of DS can be applied to the BE class, if a IP ToS precedence marking is supported. In addition to the queues that apply to each of the different services, Fig. 7 also shows the Layer 2 (L2) queue that is located on the outgoing interface card of the router. This queue is required to allow for a smooth operation of the interface and has to be configured carefully, since it can add further delay and delay jitter to all DS classes [5].

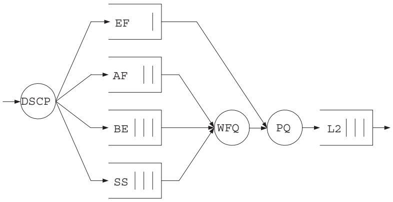


Fig. 7. DS Queuing and Scheduling

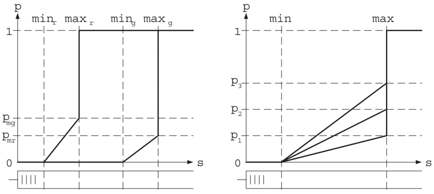


Fig. 8. MRED Configurations for AF and BE

4 Experimental Studies

We report on experiments made with a video application that utilizes different QoS classes for the transmission of layered video streams within a DS testbed. The option of SNR scalability of the H.263+ tmn encoder version 3.2 is used to generate only two layers, in order to minimize the overhead and because we think that a DS deployment with a limited number of traffic classes is more feasible. Our video transmission tool is capable of performing a marking of the packets, by setting the IP ToS or the DSCP field. Alternatively a marking according to [19] can be made in the RTP header, or by dividing the stream into sub-streams with different port fields, whereas both require a corresponding setting of the DSCP at the DS domains ingress node. In addition to the video transmission tool, we make use of the common UDP BE traffic generator gen-send/gen-recv to create congestion, and the TCP traffic generator ttcp to evaluate effects on TCP congestion control. Our testbed shown in Fig. 9 consists of four CISCO

7200 routers connected either by OC3 ATM, or Gigabit Ethernet links, whereas end systems are connected by switched Fast Ethernet. One of the ATM links is configured as a bottleneck link to 60 Mb/s, which corresponds to a netto rate of about 48 Mb/s after subtracting the ATM overhead.

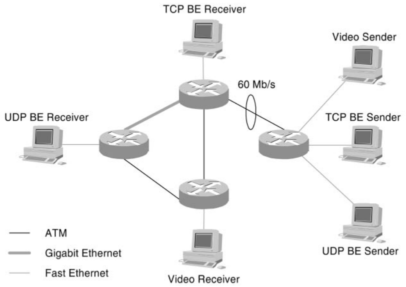


Fig. 9. DS Testbed in Jülich, donated by Cisco Systems.

The router configuration used in the tests implements an EF class by PQ, an AF class with a WFQ share of 5 %, the SS with 1 %, and the BE class with the remaining capacity. The AF class supports green and red marked packets with an M-RED implementation applying the thresholds $min_r = 16$, $max_r = 32$, $min_g = 48$, and $max_g = 64$, and the maximum probabilities $p_{mr} = p_{mg} = 0.1$. In case of pure BE experiments, a setting of the IP ToS precedence field is mapped onto an M-RED implementation with $min = 64$, $max = 128$, $p_1 = 0$, $p_2 = 0.5$, and $p_3 = 1$, which supports a high, medium, and low drop precedence. The L2 queue is set to 4 Maximum Transmission Units (MTU), here $4 \cdot 1500$ B.

We made experiments with a mapping of video layers to traffic classes according to Tab. 1. Simple BE experiments have been made as a comparison. No differentiation between EF and AF is made in scenario 5 and 6, since both classes can be used to build the guaranteed rate service that is applied here, whereas significant differences exist in terms of delay and delay jitter [23].

Table 1. Mapping of Video Layers to Traffic Classes

Sc.	Type	Dropping	Base Layer	Enhancement Layer
1	BE	Uniform	Medium Precedence	
2	DS	Uniform	AF	
3	BE	Priority	High Precedence	Low Precedence
4	DS	Priority	AF Green	AF Red
5	DS	Priority	EF/AF	BE
6	DS	Priority	EF/AF	SS

The PSNR of a repeated transmission of the "apple" sequence obtained from measurements in the testbed is shown for the different scenarios of Tab. 1 in Fig. 10–15, whereas BE congestion is created by a bursty and non-responsive 47 Mb/s UDP flow that starts after the transmission of 140 frames for the duration of 280 frames. Figure 10 shows the transmission of only the base layer in the BE class. During congestion base layer frames are dropped and the PSNR of the received stream oscillates noticeable. Thus, the quality at the receiver is degraded both, by the lower PSNR and also by the fluctuations of the PSNR, which are disturbing for the human visual system. The same applies if the sender transmits the base and the enhancement layer in the BE class as shown in Fig. 11, apart from the higher PSNR during periods without congestion. Similar experiments have been carried out with the AF implementation. The main difference is an AF reservation for the base layer. This reservation, as long as it is not exceeded, ensures the transmission of the base layer even during BE congestion as shown in Fig. 12. If the sender transmits base and enhancement layer, the AF class is oversubscribed and AF packet loss occurs during congestion. This results in similar effects as in the BE class as can be seen in Fig. 13.

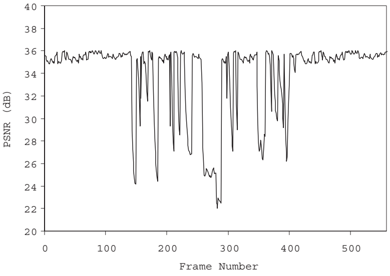


Fig. 10. Base Lay. – Medium Prec.

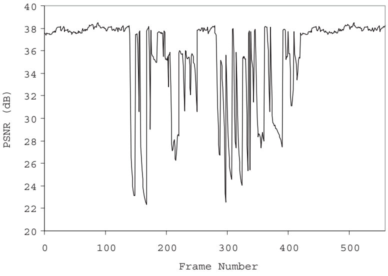


Fig. 11. Base/Enh. Lay. – Medium Prec.

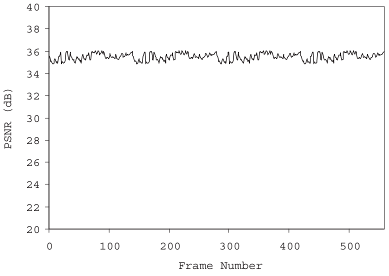


Fig. 12. Base Lay. – AF

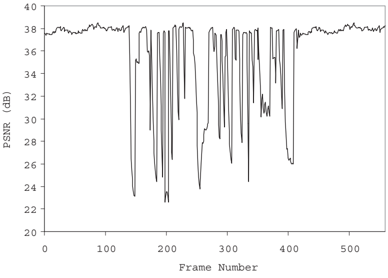
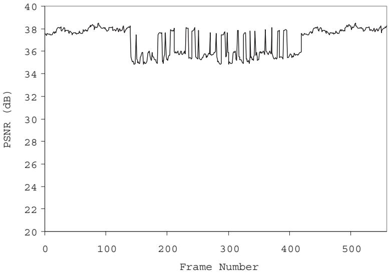


Fig. 13. Base/Enh. Lay. – AF

Figure 14 shows results from the BE video transmission, but now with a mapping of the base layer to a high and the enhancement layer to a low drop precedence. Congestion is created by a stream with medium drop precedence. In this scenario congestion only affects the enhancement layer. Base layer frames are not dropped and thus ensure a certain PSNR. Further on the PSNR oscillations are limited to a range between the PSNR of the base and the PSNR of the enhancement layer. Figure 15 shows similar results from a transmission of base and enhancement layer in the AF class with an explicit marking of the drop precedence. Again the base layer transmission can be secured and congestion only affects the enhancment layer. The PSNR measurements obtained from scenario 5 and 6 are omitted here, because they merely show similar results as given in Fig. 15. In all of the DS experiments a sufficient amount of EF or AF capacity is reserved for a loss-free base layer transmission, whereas the transmission of the enhancement layer is affected by BE congestion.



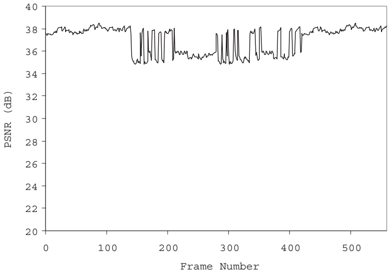


Fig. 14. Base/Enh. Lay. – High/Low Prec.

Fig. 15. Base/Enh. Lay. – AF Green/Red

The effects of the video transmission on BE TCP throughput are shown in Fig. 16–19 for the respective scenarios. For these experiments a traffic mix consisting of a non-responsive 40 Mb/s UDP flow, a TCP stream limited by the application to 12 Mb/s, and either only the base layer or base and enhancement layer of the video streams each of about 2.4 Mb/s is applied. The TCP stream is rate-limited by the Round-Trip-Time (RTT) and a maximum TCP window size configured by setting the socket buffer to 16 kB, whereas during periods of 15 s of the video base layer transmission at 10 s and the transmission of base and enhancement layer at 40 s packet loss and congestion avoidance limit TCP throughput. The base layer transmission reduces the TCP throughput by about 2.4 Mb/s, independent from the scenario and the existence of a base layer reservation, only due to TCP congestion control. The same is true for an additional enhancement layer transmission in the BE class as shown in Fig. 16, and 18. In contrast, if the AF class is oversubscribed and used for the base and enhancement layer transmission or in case of an enhancement layer transmission applying the SS, the enhancement layer stream cannot preempt the TCP stream further, as shown in Fig. 17, and 19. Thus these implementation options allow for some fairness and co-existence with responsive streams.

It has to be noted that a SS applied for delay sensitive but loss tolerant RTP/UDP traffic has to fulfill different requirements than a SS for TCP traffic, which in the first place has to avoid packet loss by offering sufficient queuing

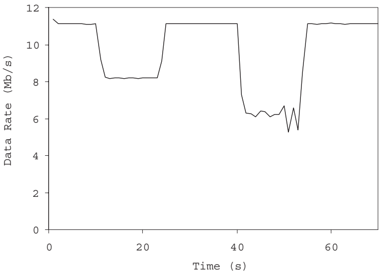


Fig. 16. BE Video vs. BE TCP

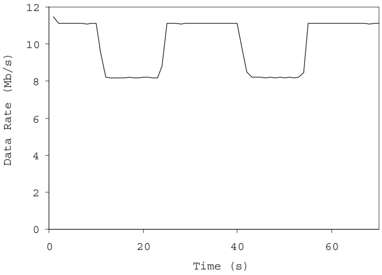


Fig. 17. AF Video vs. BE TCP

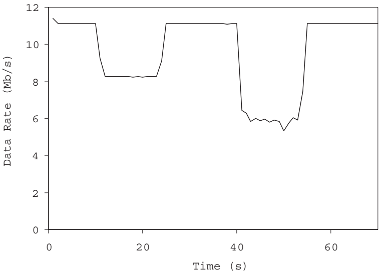


Fig. 18. EF/BE Video vs. BE TCP

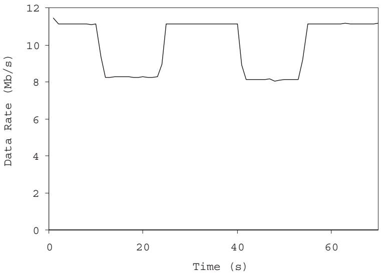


Fig. 19. EF/SS Video vs. BE TCP

space, adapted to the TCP window size. For RTP/UDP traffic rather small queues and ideally front drop have to be used, to prevent from queuing delay during periods of congestion, especially since the SS WFQ share is configured to 1 %. The situation is different for the AF M-RED implementation. The queuing delay that can be added by enhancement layer data, if mapped to the highest drop precedence level, can be configured for a single node by the setting of max_r, for a certain AF WFQ share. Further on AF traffic can be limited at DS ingress nodes and the impacts on BE TCP streams can be controlled by the assigned WFQ share, wherefore we recommend the AF based implementation.

5 Conclusions

In this paper we have shown how hierarchical video transmissions can be supported by priority dropping schemes. We have addressed the options of the H.263+ standard to generate such streams and we have shown several possible implementations of different drop probabilities in the future Internet, by applying either a BE precedence marking or the different PHB of DS. We have performed a thorough evaluation of our implementation of the proposed framework. Significant performance benefits and also the impacts on responsive flows have been shown. The different AF drop precedence level have been identified as a prime implementation option. AF can be configured to avoid base layer packet loss and it allows for applying unused capacity for an enhancement layer transmission without preempting BE TCP streams in case of congestion.

References

1. Allman, M., Paxson, V., Stevens, W.: TCP Congestion Control. RFC 2581 (1999)
2. Bajaj, S., Breslau, L., Shenker, S.: Uniform versus Priority Dropping for Layered Video. Proceedings of ACM SIGCOMM (1998)
3. Blake, S., Black, D., Carlson, M., Davies, M., Wang, Z., Weiss, W.: An architecture for Differentiated Services. RFC 2475 (1998)
4. Côté, G., Erol, B., Gallant, M.: H.263+: Video Coding at Low Bit Rates. IEEE Transactions on Circuits and Systems for Video Technology, vol. 8, no. 7 (1998)
5. Ferrari, T., Chimento, P.: A Measurement-based Analysis of Expedited Forwarding PHB Mechanisms. Proceedings of IWQoS (2000)
6. Fidler, M.: Transmission of Layered Video Streams in a Differentiated Services Network. Proceedings of AI PDCN (2002)
7. Floyd, S., Jacobson, V.: Random Early Detection Gateways for Congestion Avoidance. IEEE/ACM Transactions on Networking, vol. 1, no. 4 (1993), pp. 397-413
8. Handley, M., Padhye, J., Floyd, S.: TCP Friendly Rate Control (TFRC): Protocol Specification. Internet Draft draft-ietf-tsvwg-tfrc-03.txt (2001)
9. Heinanen, J., Finland, D., Baker, F., Weiss, W., Wroclawski, J.: An Assured Forwarding PHB. RFC 2597 (1999)
10. Heinanen, J., Guerin, R.: A single rate three color marker. RFC 2697 (1999)
11. Heinanen, J., Guerin, R.: A two rate three color marker. RFC 2698, (1999)
12. Jacobson, V., Nichols, K., Poduri, K.: An Expedited Forwarding PHB. RFC 2598, (1999)
13. Koenen, R. (Editor): Overview of the MPEG-4 Standard. ISO/IEC JTC1/SC29/WG11 (2001)
14. Kurose, J., Ross, K.: Computer Networking - A Top-Down Approach Featuring the Internet. Addison Wesley (2001) p. 80
15. Leon, D., Varsa, V.: RTP retransmission framework. Internet Draft draft-ietf-avt-rtp-selret-03 (2001)
16. McCanne, S., Jacobson, V., Vetterli, M.: Receiver-driven Layered Multicast. Proceedings of ACM SIGCOMM (1996)
17. McDysan, D.: QoS & Traffic Management in IP & ATM. McGraw-Hill (2000)
18. Miyazaki, A. et al.: RTP Payload Formats to Enable Multiple Selective Retransmission. Internet Draft draft-ietf-avt-rtp-selret-03 (2001)
19. Polk, J.: RTP Header Extension for Communications Resource Priority. Internet Draft draft-polk-avt-rtpext-res-pri-00 (2001)
20. Ramakrishnan, K., Floyd, S., Black, D.: The Addition of Explicit Congestion Notification (ECN) to IP. RFC 3168 (2001)
21. Rejaie, R., Handley, M., Estrin, M.: Quality Adaptation for Congestion Controlled Video Playback over the Internet. Proceedings of ACM SIGCOMM (1999)
22. Rosenberg, J., Schulzrinne, H.: An RTP Payload Format for Generic Forward Error Correction. RFC 2733 (1999)
23. Sander, V., Fidler, M.: Evaluation of a Differentiated Services based Implementation of a Premium and an Olympic Service. Submitted to IWQoS (2002)
24. Schulzrinne, H., Casner, S., Frederick, R., Jacobson, V.: RTP: A Transport Protocol for Real-Time Applications. RFC 1889 (1996)
25. Tan, W., Zakhor, A.: Video Multicast using Layered FEC and Scalable Compression. IEEE Trans. on Circuits and Systems for Video Tech., vol. 11, no. 3 (2001)
26. Teitelbaum, B.: Future Priorities for Internet2 QoS. www.internet2.edu/qos/wg/papers/qosFuture01.pdf (2001)
27. Wang, X., Schulzrinne, H.: Comparison of Adaptive Internet Multimedia Applications. IEICE Transactions on Communications June (1999)

Optimal Feedback for Quality Source-Adaptive Schemes in Multicast Multi-layered Video Environments*

Paulo André da Silva Gonçalves[1,2], José Ferreira de Rezende[2],
Otto Carlos Muniz Bandeira Duarte[2], and Guy Pujolle[1]

[1] LIP6 - Université Pierre et Marie Curie
8, rue du Capitaine Scott - 75015 - Paris
{Paulo.Goncalves, Guy.Pujolle}@lip6.fr
[2] GTA/COPPE - Universidade Federal do Rio de Janeiro
P.O. Box 68504 - 21945-970 - Rio de Janeiro - RJ - Brazil
{rezende, otto}@gta.ufrj.br

Abstract. Current quality source-adaptive schemes for multicast multi-layered video rely on merging capabilities at special nodes in the network as a means of combining feedback from the whole set of receivers. These strategies reduce network load and avoid feedback implosion at the source. In this paper, we examine how to provide optimal feedback for such schemes. The optimal feedback is achieved when state information from the whole set of receivers is represented in every incoming feedback packet at the source. We show that the choice of a suitable merging time window in the intermediate nodes coupled with a periodical transmission of feedback packets by the receivers leads to near-optimal feedback.

1 Introduction

The need for meeting the Quality of Service (QoS) requirements of the multimedia applications is steadily increasing. As a consequence, new approaches such as adaptive applications [1], QoS routing [2], [3] and Differentiated Services [4], [5] are currently being developed. In the case of adaptive applications, different schemes are emerging to improve quality and fairness of multi-layered video applications [6], [7], [8], [9], [10], [11], [12], [13], [14]. In the context of unicast, as an example, Rejaie et al. [8], [9] propose an adaptation scheme in which the perceived video quality at the receiver is more stable even in the presence of fluctuations in the available network bandwidth. In this scheme, the receiver plays an important role by buffering video layers and sending to the source feedback packets containing current network congestion information.

In the context of multicast, exchanging network congestion information or more generally any kind of information between receivers and source is a nontrivial matter. The challenge facing this approach is twofold. First, the source

* This work was sponsored by: CAPES/Brazil, COFECUB, CNRS, and CNPq. Paulo André da Silva Gonçalves has a scolarship from CAPES/Brazil.

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 563–574, 2002.

will be prone to feedback implosion. Second, even if feedback implosion avoidance mechanisms are employed, a means of allowing that the incoming feedback at the source represents the current state of the whole set of receivers should be provided. In the contrary, i.e. in the case of the source has only a partial state information, the source adaptation may be unsuitable and unfair when regarding receivers whose state is not included in feedback packets. If every incoming feedback at the source represents the state information from the whole set of receivers in a multicast group, *optimal feedback* is achieved.

In this paper, we investigate how to provide *optimal feedback* for current quality source-adaptive schemes in multicast multi-layered video environments. The efficiency of such schemes in terms of quality/fairness, network load and source implosion depends on the choice of three different policies: feedback transmission, temporal-merging and content-merging policies. The first policy is related to *when* receivers should send feedback packets to the source. The second one determines *when* feedback packets are merged by intermediate nodes. The last policy establishes *how* feedback packets are merged. The effectiveness of the feedback transmission and temporal-merging policies, which are investigated in this paper, dictates the efficiency of the third policy.

The paper is organized as follows. In Section 2, we explain the motivation for this work by presenting current quality source-adaptive schemes proposed in the literature. In Section 3, we describe the temporal-merging policy and analyze in which conditions this policy provides *optimal feedback*. The formulas that model the temporal-merging policy are presented in Section 4. We thereafter analyze how *optimal feedback* is related to feedback suppression. In Section 5, we provide a feedback transmission policy that coupled with the temporal-merging policy leads to an optimal feedback. Theoretical and simulation results are presented in Section 6. Finally, in Section 7 we conclude the paper.

2 Motivation

The Receiver-driven Layered Multicast (RLM) [6] scheme pioneered the study of the multicast multi-layered video transmission. In RLM, each video layer is sent out on a different multicast group. Based on congestion control information, receivers can subscribe to a number of layers that can be supported by their individual path from the source. The Layered Video Multicast with Retransmissions (LVMR) [7] employs a hierarchical rate control to manage the adding and dropping of video layers by receivers. Furthermore, LVMR deploys an error recovery mechanism using retransmissions to adapt to network congestion. Despite the quality adaptation being provided by the schemes cited above, receivers are limited to the layers the source decides to transmit. Current quality source-adaptive schemes provide different solutions to overcome this limitation.

In [11], Vickers et al. propose a source-adaptive multi-layered multicast algorithm in which feedback control packets containing current network congestion information are exchanged between source and receivers. In this algorithm, the source periodically sends to the multicast group a control packet that is updated with congestion control information as it traverses the branches of the multicast tree. Upon receiving a control packet, the receiver returns to the source a feedback packet containing the received congestion control information. Based on the goodput quality metric, intermediate nodes combine feedback packets' contents in order to estimate the number of video layers and their respective transmission rates. Another scheme for adapting the quality of multi-layered video is proposed in [15]. The control information at intermediate nodes is similar to the approach in [11]. However, different from other proposals, content-merging procedures are performed in a single loop and the concept of virtual layers is employed. The result is an improvement in the fairness of the delivered service.

The classical approach to compute the rates of the layers for source-adaptive multi-layered multicast schemes is based on the goodput quality metric. This approach is analyzed in [13]. The authors have demonstrated that the classical approach fails in the computation of the most adequate rates for the layers. In another work [10], the same authors propose another scheme for quality adaptation. Such a scheme is based on a combined metric that allows for the density of satisfied users, the weighted bandwidth allocation, and the goodput quality at the receivers. The proposed scheme yields enhancements in quality/fairness of multi-layered multicast sessions.

The quality source-adaptive schemes we mentioned earlier rely on the same temporal-merging policy performed at intermediate nodes. However, such schemes focus on how to efficiently combine packets' contents to yield improvements in the delivered service. We argue that the efficiency of such schemes can be maximized if *optimal feedback* is provided.

3 Temporal-Merging Policy

Consider our basic network topology as shows Fig. 1. The feedback-merger capable-node waits for packets from k receivers. The temporal-merging policy is performed as follows. Incoming feedback packets are merged and forwarded upstream if one of the following two *conditions* holds. *(i)* At least one feedback packet from each node has arrived. *(ii)* The merger timer expires. If more than one feedback packet from the same node arrives before merging, the previous stored packet is replaced.

The basic strategy to achieve an optimal feedback is to maximize the number of packet merges. In this way, as regarding the temporal-merging policy described above, we should provide a means of minimizing the number of packet replacements and maximizing the number of packet merges due to *condition (i)*. Providing such a means can ultimately be regarded as finding a feedback transmission policy and a merging time window that will allow to achieve an optimal feedback.

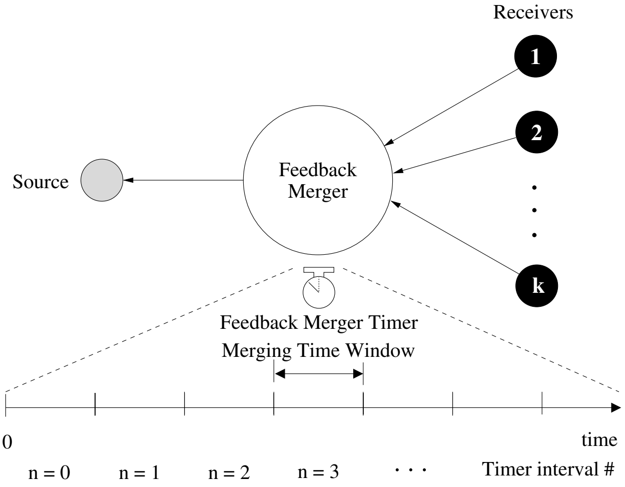


Fig. 1. Basic network topology and merger timer schematic

4 Temporal-Merging Policy Model

In this section, we first describe a formula to compute the number of merges due to *conditions (i)* and *(ii)*, and next we analyze how *optimal feedback* is related to feedback suppression. In our analysis, we consider that the merger timer expires in a periodical fashion. Another possible approach would be to reschedule the timer upon arriving the first candidate packet for merging. Despite this approach only implies changes in the conditions for which formulas presented throughout this section hold, the effects of such approach are not evaluated in this paper.

4.1 Merges Due to *Conditions (i)* and *(ii)*

Consider the merger timer schematic shown in Fig. 1. For a given timer interval n, let ρ_i^n and β_i^n be, respectively, the number of packets arrivals from node i and the number of packet replacements for nodes i. Then, for a feedback merger waiting for packets from k nodes, the number of outcoming and incoming packets in the timer interval n is, respectively, given by

$$\gamma_{out}^n = \max_{i=1\ldots k}[\rho_i^n - \beta_i^n] \tag{1}$$

and

$$\gamma_{in}^n = \sum_{i=1}^{k} \rho_i^n \ . \tag{2}$$

Given a timer interval n, the number of outcoming packets due to *condition (i)* and *condition (ii)* is, respectively, given by

$$\mu_k^n = \min_{i=1\ldots k}[\rho_i^n - \beta_i^n] \tag{3}$$

and

$$\sigma_k^n = \max_{i=1\dots k}[\rho_i^n - \beta_i^n] - \min_{i=1\dots k}[\rho_i^n - \beta_i^n] \ . \tag{4}$$

If upon expiring the merging timer there exists at least one feedback packet waiting for merging then $\sigma_k^n = 1$ else $\sigma_k^n = 0$. Note that the number of outcoming packets in any timer interval n is equal to $\mu_k^n + \sigma_k^n$ that provides the same result as in (1).

Considering N timer intervals, the ratio of outcoming packets due to each *condition* to the number of incoming packets is, respectively, given by

$$\theta_k^{cond(i)} = \frac{\sum_{n=0}^{N-1} \mu_k^n}{\sum_{n=0}^{N-1} \gamma_{in}^n} \qquad 0 \le \theta_k^{cond(i)} \le 1 \ , \tag{5}$$

and

$$\theta_k^{cond(ii)} = \frac{\sum_{n=0}^{N-1} \sigma_k^n}{\sum_{n=0}^{N-1} \gamma_{in}^n} \qquad 0 \le \theta_k^{cond(ii)} \le 1 \ . \tag{6}$$

These formulas hold if for some $0 \le n \le (N-1)$ and $1 < i \le k$, there exists a γ_{in}^n such that $\gamma_{in}^n \ge 1$. The fraction of merges due to *condition (i)* and *condition (ii)* is, respectively, given by

$$\alpha_k^{cond(i)} = \frac{\sum_{n=0}^{N-1} \min_{i=1\dots k}[\rho_i^n - \beta_i^n]}{\sum_{n=0}^{N-1} \max_{i=1\dots k}[\rho_i^n - \beta_i^n]} \qquad 0 \le \alpha_k^{cond(i)} \le 1 \ , \tag{7}$$

and

$$\alpha_k^{cond(ii)} = 1 - \frac{\sum_{n=0}^{N-1} \min_{i=1\dots k}[\rho_i^n - \beta_i^n]}{\sum_{n=0}^{N-1} \max_{i=1\dots k}[\rho_i^n - \beta_i^n]} \qquad 0 \le \alpha_k^{cond(ii)} \le 1 \ . \tag{8}$$

These formulas hold if for some $0 \le n \le (N-1)$ and $1 < i \le k$, there exists a ρ_i^n such that $\rho_i^n \ge 1$. In Section 6 we will use these formulas to find a merging time window that minimizes the fraction of merges due to *condition (ii)*. Since $\alpha_k^{cond(ii)}$ is the complementary value of $\alpha_k^{cond(i)}$, the minimization of the fraction of merges due to *condition (ii)* implies the maximization of the fraction of merges due to *condition (i)*. From this fact, minimization of $\alpha_k^{cond(ii)}$ and maximization of $\alpha_k^{cond(i)}$ are interchangeable throughout this paper.

4.2 Feedback Suppression

The notion of a high level of feedback suppression is commonly employed as a measure of efficiency for implosion avoidance. Let us then verify how *optimal feedback* is related to feedback suppression. We define the *Filtering Level* (ξ^n) in the timer interval n as the number of packets suppressed by the feedback merger over the total number of incoming packets, that is,

$$\xi^n = 1 - \frac{\gamma_{out}^n}{\gamma_{in}^n} \qquad 0 \le \xi^n < 1 \ , \tag{9}$$

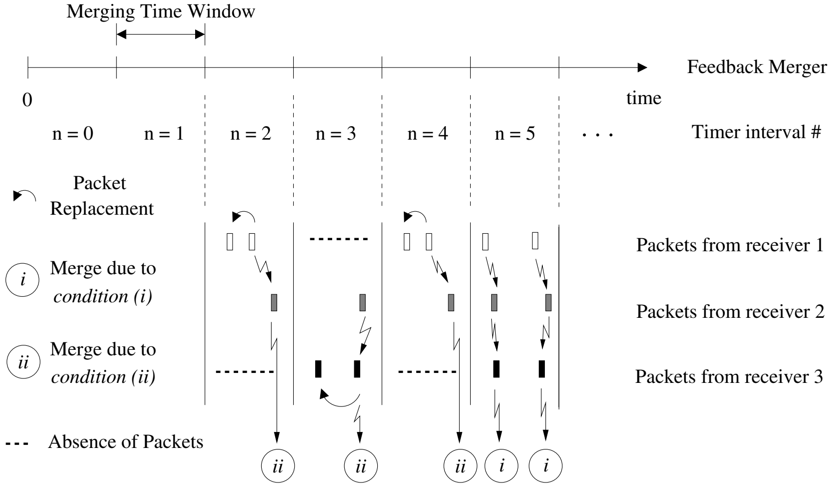


Fig. 2. Scenario in which the *Global Filtering Level* assumes its optimal value

if $\gamma_{in}^n \geq 1$. In generalizing our definition to take into account a number N of timer intervals, we get the *Global Filtering Level*

$$\xi_k = 1 - \frac{\sum_{n=0}^{N-1} \gamma_{out}^n}{\sum_{n=0}^{N-1} \gamma_{in}^n} \qquad 0 \leq \xi_k < 1 \ . \tag{10}$$

Substituting equations (1) and (2) in (10), we get

$$\xi_k = 1 - \frac{\sum_{n=0}^{N-1} \max_{i=1 \ldots k}[\rho_i^n - \beta_i^n]}{\sum_{n=0}^{N-1} \sum_{i=1}^{k} \rho_i^n} \qquad \text{for } k > 1 \ , \tag{11}$$

and if for some $0 \leq n \leq (N-1)$ and $1 \leq i \leq k$, there exists a ρ_i^n such that $\rho_i^n \geq 1$. If $k \leq 1$ then $\xi_k = 0$.

Now let us find the *optimal value* for ξ_k. For this purpose, consider both the network example as depicted in Fig. 1 and the following scenario where *optimal feedback* occurs. For every timer interval n having packets within there are no packet replacements and the number of packet arrivals from all nodes is the same. Thus, for any timer interval n having packets within, the ratio $\gamma_{out}^n/\gamma_{in}^n$ will always be constant and equal to $1/k$. Without loss of generality, the ratio $\sum_{n=0}^{N-1} \gamma_{out}^n / \sum_{n=0}^{N-1} \gamma_{in}^n$ will also be equal to $1/k$. In this manner, the *optimal value* for ξ_k is given by

$$\xi_{k,optimal} = 1 - \frac{1}{k} \qquad \text{for } k > 1 \ . \tag{12}$$

A necessary condition to achieve an optimal feedback is that the *Global Filtering Level* be as close as possible of its optimal value. However, an optimal value of the *Global Filtering Level* does not imply *optimal feedback* because packet replacements may be occurring as well as merges due to *condition (ii).*

This situation is illustrated in Fig. 2. Considering the timer intervals from 2 to 5, the *Global Filtering Level* (ξ_3) is equals to its optimal value of 0.66.

Other important remarks are the following. If there are no packet replacements, ξ_k will be less or equal to $\xi_{k,optimal}$. If, on the other hand, there are packet replacements, ξ_k may assume values greater than $\xi_{k,optimal}$. If $\alpha_k^{cond(ii)}$ is equal to 0 then ξ_k will be equal or greater than $\xi_{k,optimal}$. In this case, ξ_k will be equal to $\xi_{k,optimal}$ if there are no packet replacements and it will be greater than $\xi_{k,optimal}$ if there is at least one packet replacement. Thus, despite a *Global Filtering Level* (ξ_k) greater than its optimal value impacts better the efficiency for feedback implosion avoidance, it is not adequate to an effective feedback.

5 Choice of the Feedback Transmission Policy

Both the number of packets and the number of packet replacements in each timer interval depend on the feedback transmission policy. When a number of feedback packets must be sent from receivers to the source, packet replacements can be minimized if we allow the arrival rate of feedback from the different receivers at a feedback merger be the same. In practice, it is a nontrivial matter to ensure that the arrival rate of feedback be the same because feedback packets may suffer from random delays into the network and be lost. However, a first approximation is to assign to all receivers the same feedback sending interval.

Since we defined a feedback transmission policy, let us now provide means of quantifying ρ_i^n. For this purpose, recall our basic network topology and consider both schematics the feedback transmission and the merger timer expiration as depicted in Fig. 3. The merger timer expires in a periodical fashion at time instants $\tau_{exp}^{(n)}$ given by (13), where $n = 0, 1, 2, \ldots$ is the current timer interval, $\Delta\tau$ is the merging time window size, and $\delta^{(n)}$ is a random variable representing jitter in the timer interval n.

$$\tau_{exp}^{(n)} = \begin{cases} \Delta\tau + \delta^{(n)} & \text{if } n = 0 \\ \tau_{exp}^{(n-1)} + \Delta\tau + \delta^{(n)} & \text{otherwise .} \end{cases} \tag{13}$$

As depicted in Fig. 3, each receiver i transmits feedback packets in a periodical fashion. The arrival time $\tau_{arrival,i}^{(r_i)}$ of a feedback packet at the feedback merger is given by (14), where $r_i = 0, 1, 2, \ldots$ is the current sending round number for receiver i, Δt represents the feedback sending interval and $\sigma_i^{(r_i)}$ models processing time delay in the sending round r_i. The start-time of a receiver i is denoted by c_i. Queuing delay in a given sending round r_i, transmission delay and propagation delay are, respectively, denoted by $Q_i^{(r_i)}$, T_i and P_i.

$$\tau_{arrival,i}^{(r_i)} = \begin{cases} c_i + Q_i^{(r_i)} + P_i + T_i & \text{if } r_i = 0 \\ \tau_{arrival,i}^{(r_i-1)} + \Delta t_i + \sigma_i^{(r_i)} + Q_i^{(r_i)} + P_i + T_i & \text{otherwise .} \end{cases} \tag{14}$$

In (14), we assume that there is no packet loss and the transmission buffer at nodes is infinite. Feedback packets that are candidates to be merged must arrive

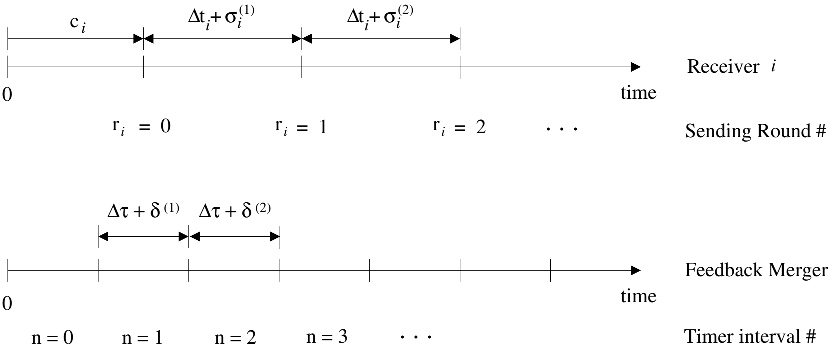


Fig. 3. Feedback transmission policy and merger timer expiration schematics

at the feedback merger between two consecutive timer expirations. Thus, based on (13) and (14) we can write the following.

$$h\tau_{exp}^{(n-h)} \leq \tau_{arrival,i}^{(r_i-m)} + m(\varDelta t_i + \sigma_i^{(r_i)} + Q_i^{(r_i)} + P_i + T_i) < \tau_{exp}^{(nh)} \ , \qquad (15)$$

where m is equal to 0 if $r_i = 0$ and equal to 1 otherwise and h is equal to 0 if $n = 0$ and equal to 1 otherwise. For a given timer interval n, let $\varUpsilon_i^n$ be the set of elements $r_i \in \mathbb{N}$ for which the inequalities above are respected. Now let $\mathfrak{N}(\varUpsilon_i^n)$ be the number of elements in $\varUpsilon_i^n$. Then, the number of packets from node i in any timer interval n is $\rho_i^r = \mathfrak{N}(\varUpsilon_i^n)$.

6 Results

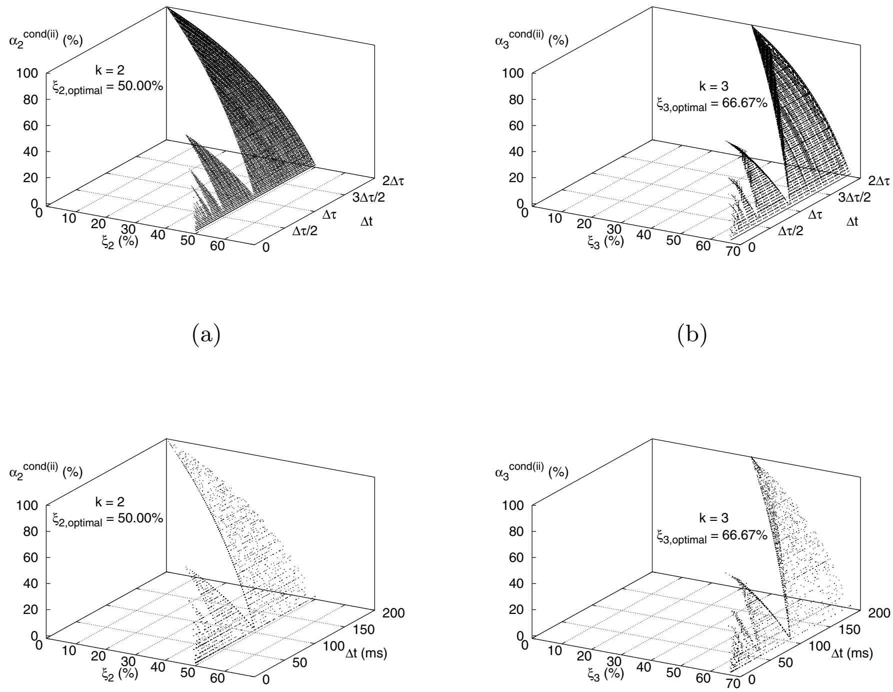
In this section we present our theoretical and simulation results.

6.1 Theoretical

We assume that feedback packets from different receivers arrive at the feedback merger on different incoming links. A contrary assumption only impacts the minimal merging time window value that can be chosen. Finding such a value is outside of the scope of this paper. We set $\delta^{(n)}$, $\sigma_i^{(r_i)}$, and $Q_i^{(r_i)}$ to zero in order to assess the isolated behavior of the temporal-merging policy coupled with our choosen feedback transmission policy. In this manner, we will be able to find a merge time window that ensures 100% of merges due to *condition (i)* at steady state.

Respectively for $k = 2$ and $k = 3$, Figs. 4(a) and 4(b) show the behavior at steady state of the pair $(\xi_k, \alpha_k^{cond(ii)})$ as both the feedback sending interval and the deviation among receivers' start-time vary. Note that for a given $\varDelta t$, different pairs $(\xi_k, \alpha_k^{cond(ii)})$ come from the different deviations among receivers' start-time. Packet replacements were not observed. The main results are the following.

First, merges due to *condition (ii)* lead to a *Global Filtering Level* less than its optimal value, and second, assigning the merging time window to a multiple of the feedback sending interval ensures 100% of merges due to *condition (i)*, regardless of receivers' start-time. That is, packet merges are maximized for $\varDelta\tau = b\varDelta t$, where $b = 1, 2, \ldots$, and as a consequence, *optimal feedback* is provided to the source. Since the maximum waiting time for merging feedback packets is directly proportional to the factor b, we adopt its minimum value, i.e. $b = 1$.



(a)

(b)

(c)

(d)

Fig. 4. Theoretical and simulation results

6.2 Simulation

We validate our theoretical results through simulation using ns-2 [16], [17]. Figs. 4(c) and 4(d) show the results we obtained under the same conditions earlier described in this section. Let us now verify how the previous results are impacted in a heterogeneous network with unpredictable random delays. Fig. 5

shows the transit-stub network topology used in our simulations. Link capacities are 10 Mbps. Each link delay was randomly chosen uniformly on the interval [1, 10] ms. All intermediate nodes are capable of merging feedback packets. The merging time window and the feedback sending interval were set to 500 ms. Background traffic is generated on the transit domain throughout all simulation time. For each sending round, processing delay at receivers was uniformly chosen at random on the interval [0, 5] ms. Merging time window variation in each merging procedure was randomly chosen uniformly on the interval [0, 5] ms. The start-time of each receiver was randomly chosen uniformly on the interval [0, 1] s. Each simulation was run for 315 seconds. All simulation results have a confidence interval of 90%.

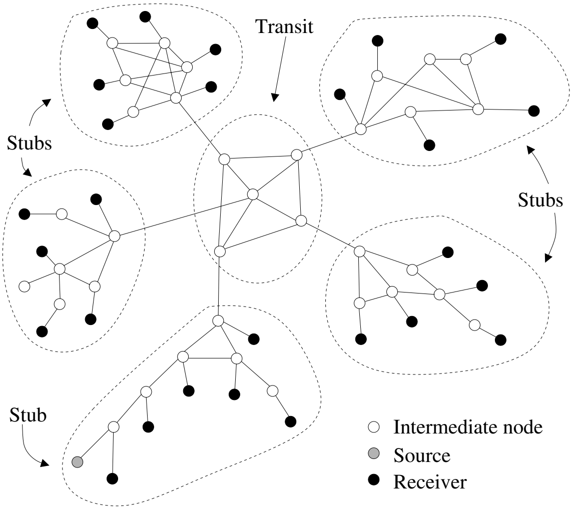
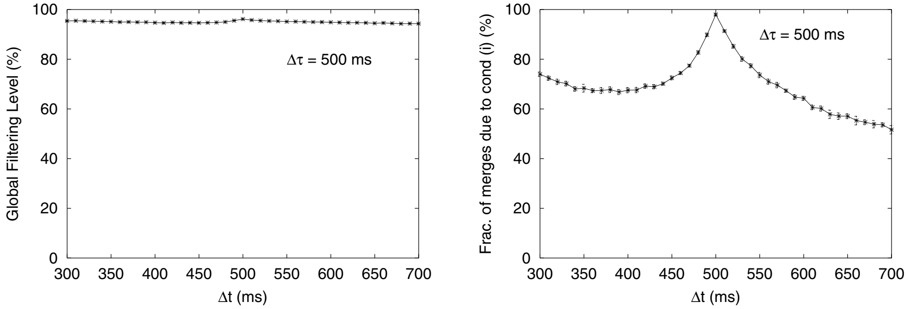


Fig. 5. Simulation Topology

The *Global Filtering Level* and the fraction of merges due to *condition (i)* were analyzed varying the feedback sending interval. Packet replacements were less than 0.4% for the feedback sending interval equals to the merging time window size. As regarding the others feedback sending intervals up to 5% of packet replacements were observed.

Fig. 6(a) shows a *Global Filtering Level* varying from 94.30% to 96.16%, making evident the effectiveness of the temporal-merging policy for feedback implosion avoidance. The *Global Filtering Level* achieves a maximum value of 96.16% when the merging time window size is assigned to the value of the feedback sending interval. In this case, the *Global Filtering Level* tended to its optimal value. For the others feedback sending intervals, the *Global Filtering Level* did not exceed this maximum value. That comes from the following fact. If on the one hand merges due to *condition (ii)* lead to a *Global Filtering Level* less than its optimal value, on the other hand, packet replacements lead to a *Global Filtering Level* greater than its optimal value. Thus, merges due to *condition (ii)* had a greater impact on the *Global Filtering Level* than packet replacements.



(a) (b)

Fig. 6. *Global Filtering Level* and fraction of merges due to *condition (i)*

Fig. 6(b) shows how the merges occur into the network as the feedback sending interval is varied. In particular, around 2% of merges occur due to *condition (ii)* when the merging time window is assigned to the value of the feedback sending interval. This result shows that unpredictable random delays negatively impacts the fraction of merges due to *condition (i)*, and as a consequence, only *near-optimal feedback* can be provided to the source. Nevertheless, this negative effect can be minimized if feedback mergers employ an adaptive merging time window.

7 Conclusion

In this paper, we have addressed the issue of providing *optimal feedback* for quality source-adaptive schemes in multicast multi-layered video environments. We argued that while current quality source-adaptive schemes focus on how to combine efficiently packets' contents to yield improvements in the delivered service, they are incapable of achieving their maximal efficiency if *optimal feedback* is not provided. We investigated how *optimal feedback* is related to feedback suppression. Our analysis showed that there exists an optimal value for the packet filtering level. Higher values lead to non-optimal feedback despite they imply efficiency for feedback implosion avoidance. We also demonstrated that by sending feedback packets in a periodical fashion and assigning the merging time window to the sending interval of feedback, *near-optimal feedback* can be achieved.

References

1. C. Diot, C. Huitema, and T. Turletti, “Multimedia applications should be adaptive,” in *HPCS’95*, (Mystic, CN), Aug. 1995.
2. Z. Wang and J. Crowcroft, “Quality of service routing for supporting multimedia applications,” *IEEE Journal on Selected Areas in Communications*, vol. 14, Sept. 1996.
3. L. H. M. K. Costa, S. Fdida, and O. C. M. B. Duarte, “A scalable algorithm for link-state QoS-based routing with three metrics,” in *IEEE ICC’2001*, (Helsink, Filand), June 2001.
4. S. Blake *et al.*, *An Architecture for Differentiated Services.* Internet RFC 2475, Dec. 1998.
5. A. Ziviani, J. F. de Rezende, O. C. M. B. Duarte, and S. Fdida, “Evaluating voice traffic in a differentiated services environment,” in *Proceedings of the 17th International Teletraffic Congress - ITC17*, (Salvador, Brazil), Dec. 2001.
6. S. McCanne, V. Jacobson, and M. Vetterli, “Receiver-driven layered multicast,” in *SIGCOMM’96*, (Stanford, CA), pp. 117–130, Aug. 1996.
7. X. Li, S. Paul, and M. H. Ammar, “Layered video multicast with retransmission (LVMR): Evaluation of hierarchical rate control,” in *IEEE INFOCOM’98*, (San Francisco, CA), pp. 1062–1072, Apr. 1998.
8. R. Rejaie, D. Estrin, and M. Handley, “Quality adaptation for congestion controlled video playback over the internet,” in *Proceedings of ACM SIGCOMM’99*, (Cambridge, MA), Sept. 1999.
9. R. Rejaie, M. Handley, and D. Estrin, “Layered quality adaptation for internet video streaming,” *IEEE Journal on Selected Areas in Communications*, vol. 18, Dec. 2000.
10. M. D. de Amorim, O. C. M. B. Duarte, and G. Pujolle, “Application-Aware multicast,” in *IEEE GLOBECOM’2001*, (San Antonio, Texas), Nov. 2001.
11. B. J. Vickers, C. Albuquerque, and T. Suda, “Adaptive multicast of multi-layered video: Rate-based and credit-based approachs,” in *IEEE INFOCOM’98*, (San Francisco, CA), Apr. 1998.
12. P. A. da Silva Gonçalves, J. F. de Rezende, and O. C. M. B. Duarte, “An active service for multicast video distribution,” *Journal of the Brazilian Computer Society*, vol. 7, pp. 43–51, July 2000.
13. M. D. de Amorim, O. C. M. B. Duarte, and G. Pujolle, “Multi-criteria arguments for improving the fairness of layered multicast applications,” in *Lecture Notes in Computer Science*, no. 1815, pp. 1–10, May 2000.
14. B. J. Vickers, M. Lee, and T. Suda, “Feedback control mechanisms for real-Time multipoint video services,” *IEEE Journal on Selected Areas in Communications*, vol. 15, pp. 512–530, Apr. 1997.
15. M. D. Amorim, O. C. M. B. Duarte, and G. Pujolle, “Single-loop packet merging for receiver oriented multicast multi-layered video,” in *International Conference in Computer Communications*, (Tokyo, Japan), Sept. 1999.
16. K. Fall and K. Varadhan, “NS notes and documentation,” tech. rep., The VINT Project, July 1999.
17. S. Bajaj *et al.*, “Improving simulation for network research,” tech. rep., University of Southern California, Department of Computer Science, Mar. 1999.

A Fibre Channel Dimensioning for a Multimedia System with Deterministic QoS

Laurent George[1], Dana Marinca[2], and Pascale Minet[3]

[1] University of Paris 12, LIIA, 120 rue Paul Armangot, 94400 Vitry sur Seine, France,
george@univ-paris12.fr

[2] University of Versailles, 45 avenue des Etats-Unis, 78035 Versailles Cedex, France
dimarinca@free.fr

[3] INRIA, Rocquencourt, BP105, 78153 Le Chesnay Cedex, France
pascale.minet@inria.fr

Abstract. We propose to use Fibre Channel (FC) technology in multimedia systems offering Video on Demand (VoD) services. The Storage Area Network (SAN) is based on (i) FC-loops connecting magnetic disks and on (ii) FC-switches connecting loops to servers. We show how to dimension FC-loops to offer a deterministic guarantee of Quality of Service to the VoD clients. The performance results of this analysis, confirmed by already published simulation results, enable to determine the optimal number of disks connected to a loop and the maximum number of clients acceptable by a loop. We study the influence of disk performance and determine the best number of blocks to retrieve per disk request.

Introduction

Multimedia systems can be used in many domains: entertainment in hotels, telelearning, production in radio/TV studios,... In this paper, we are concerned with the design of multimedia systems providing VoD (Video on Demand) to their clients. The client may interact by means of VCR commands (i.e. start/stop, pause/play, and jump backward/forward). We are interested in multimedia systems providing a deterministic guarantee of Quality of Service (QoS) to their clients. A multimedia system consists of different components in charge of storage/retrieval of multimedia data, network communication, and system activity control. Each component contributes to ensure the end-to-end QoS. In this paper, we focus on the storage system, a main component of the multimedia system. We propose to use a Storage Area Network (SAN) based on Fibre Channel (FC) technology. Indeed, FC offers a high performance environment for the communications between computers and the storage system and allows a very scalable architecture. We show how to dimension such a system based on a worst case analysis. We determine the maximum number of acceptable clients and the optimal number of disks per loop. We study the influence of disk performance, and size of the data retrieved by the disk. The worst case analysis can be used by the admission control to decide on the acceptance of a new client. If accepted, this client will receive a deterministic QoS guarantee.

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 575–587, 2002.

This paper is organized as follows. In section 1, we briefly present the components of a multimedia system and give the properties that must be achieved by such a system. In section 2, we describe the main features of Fibre Channel and a SAN architecture based on this technology. In section 3, we propose a performance analysis of an arbitrated loop connecting servers and magnetic disks. We first recall classical results for real-time non-preemptive uniprocessor scheduling. These results are then applied to disks scheduling and FC-loop access scheduling. This system behavior has been simulated ([12], [13]) and the associated results have been published. These results are used to validate our analysis. Then, we show how to use our results to dimension the storage system. Finally, we conclude.

1 Multimedia Systems

In this section we describe the general architecture of a multimedia system, defining the required properties to achieve the requested QoS.

1.1 General Architecture of a Multimedia System

A multimedia system consists of four main components: the servers, the storage system, the network and the clients. A server is in charge of transmitting multimedia contents from the storage system to the clients or from a multimedia source to the storage system. In this paper we assume that the servers access the storage system by means of a network, as illustrated by figure 1.

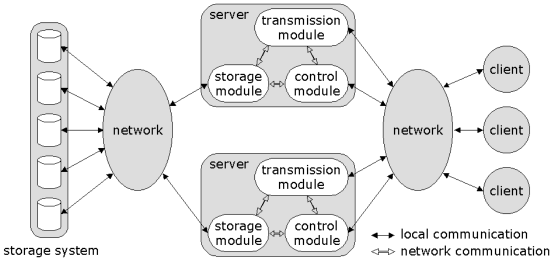


Fig. 1. General multimedia system architecture

A server consists of three modules: a control module, a storage module and a transmission module. The control module is in charge of presenting the catalog of available multimedia contents to the clients, applying the admission control to accept new clients, controling the general activity of the server. The storage module is in charge of transfering multimedia contents between storage system and main memory of the server. The transmission module transfers multimedia contents from server main memory to the clients. Furthermore, it receives the clients VCR commands controling the multimedia streams.The storage system can be constituted by magnetics disks, video tapes or CD-ROMs libraries. Because of shorter access time, we assume that the storage system is based on magnetics disks.

From the client point of view, the VoD system QoS is characterized by the following requirements:
⋆ (R1) short and upper bounded response time to VCR commands (start, stop, play, jump);
⋆ (R2) fluid visualization of any video content;
⋆ (R3) a minimum interruption of the transmission in case of failures;
⋆ (R4) a various choice of video contents.
Each component of the VoD system contributes to achieve the end-to-end QoS.

1.2 State of the Art

Our main contribution concerns the dimensioning of a SAN based on Fibre Channel in a multimedia system providing VoD services and offering a deterministic Quality of Service. As previously seen, such a system must (i) offer a low latency for VCR commands, (ii) require a small amount of buffers, avoid video starvation as well as buffer overflow, and (iii) support a large number of clients with guaranteed QoS. This goal has been expressed in a lot of papers [6], [7], [8] and [9]. The video striping policy and the scheduling policy are of prime importance to achieve this goal.

With regard to video striping, a classification has been introduced in [3], according to the striping applied to (i) the video content and (ii) on the segment. Each striping can be wide (over all disks of the server), narrow (limited to a subset of disks, for instance the disks connected to a Fibre Channel arbitrated loop) or single (one disk). The combination of a wide video striping and a single segment striping is also called Coarse Grain Striping in [4] and [5]. In the case of a VoD system based on Fibre Channel technology, magnetic disks are connected to arbitrated loops. If we consider an arbitrated loop, the best load balancing of disks connected to this loop leads to split a video content over all the disks of this loop. Moreover, as a disk must first win the loop arbitration before being authorized to transmit the requested data, the technique consisting in striping a segment over several disks loses its interest. That is why a segment is stored on a single disk. We then get the architecture described in section 2.2. The closest work to our corresponds to [3], however it only considers the disk retrieval time and does not account for the time needed to access a shared medium, a Fibre Channel loop in our case. Performance results concerning disks connected to a Fibre Channel arbitrated loop are given in [12] and [13]; these results have been obtained by simulation. We use these results to validate our model in different configurations.

2 Fibre Channel for a Multimedia System

2.1 Fibre Channel Principles

Fibre Channel, FC, is an ANSI standard defining high throughput network technology [1]. Advantages offered by the Fibre Channel technology are:

1. Flexibility. FC technology defines three physical topologies: point-to-point, switched topology also named Fabric and (3) ring topology, also named Arbitrated Loop. These topologies can use optical fibre or copper cable.
2. Performances. FC allows high speed links and several throughputs from 133 Mbps to 1026 Mbps. FC Arbitrated Loop technology allows the interconnection of 127 nodes. FC Fabric allows a maximum of 2^{24} nodes.
3. Load balancing. This technology allows concurrent accesses of servers to the same storage system.
4. Availability. The insertion of a new node can be realized without disconnecting the system. In an Arbitrated Loop topology, each node represents a single point of failure. This drawback can be eliminated by using a hub. Dual loop can be used to tolerate the failure of the medium on one loop.

In this paper, we focus on the arbitrated loop topology. Class 3 is the only possible class on an arbitrated Loop. Class 3 does not require acknowledgements. Frames are used to transfer data.The maximum payload in a data frame is 2112 bytes. To control the data transmission, FC uses control information [1]. For instance, R_RDY, receiver ready, is used for the buffer to buffer flow control, ARB is used to arbitrate the loop access, OPN is used by a Port which owns the loop to initiate a communication with another port on the Loop, and CLS used to finish the communication between two ports on a Loop.

2.2 SAN Based on Fibre Channel in a Multimedia System

A Storage Area Network, SAN, is a high-speed, scalable network of storage devices, servers (connected entities) and interconnecting entities (switch, hub). As in [2], we propose to use Fibre Channel for the SAN. We first present the adopted architecture and then describe how the video contents are stored.

- **VoD system architecture**

The architecture we propose is based on Fibre Channel Storage Area Network. The storage system is made up by magnetic disks. They are interconnected by means of one or several FC Arbitrated Loops. FC-loops are connected to servers by means of FC-switches. On the other hand, a multimedia system must allow easy extensions when necessary. This situation occurs when for instance the server has to serve a higher number of clients, or the storage capacity must be increased. A modular and flexible architecture is thus necessary. The proposed architecture meets these goals: for instance, we can connect additional arbitrated loops in the storage system without interrupting the system activity.

- **Video content storage**

We assume that the video contents are coded at a constant throughput. Any video content V is stored on all the disks of a loop L. We assume that all the disks in the system have the same block size B. The video content V is split up on all the disks of loop L by m data blocks of size B: one disk contains the first m blocks, another disk contains the m next blocks, where m is a parameter of the storage system introduced to minimize the overhead induced by the disk seek time and rotational latency. Indeed, m blocks of size B are retrieved in a single disk request. We will see in section 3.4 how to determine the best value of m. According to the classification of [3], this configuration corresponds to a narrow striping of the video content and a segment striped on a single disk.

3 Performance Analysis of an Arbitrated Loop

In this section, we establish the feasibility conditions associated with an arbitrated loop of a VoD storage system. We focus on two resources: the magnetic disks and the arbitrated loop. The feasibility conditions are based on the worst case analysis detailed in section 3.1. We show how to apply those results to model the behavior of an arbitrated loop in a SAN system. In this analysis, the feasibility conditions are established between the magnetic disks and the server. To simplify the analysis, we assume that the server connected to the FC fabric encounters a constant delay (no jitter) through the fabric.

3.1 Uniprocessor Real-Time Scheduling

We now focus on uniprocessor real-time scheduling. The results presented here are used in section 3.3, in the context of a storage system based on FC technology. First, we recall some real-time scheduling results for Non-Preemptive Fixed Priority/Highest Priority First (NP-FP/HPF) scheduling. Then we establish the feasibility conditions for sporadic tasks executed with NP-FP/HPF scheduling.

- **Concepts and notations**

We investigate the problem of scheduling a set $\tau = \{\tau_1, ...\tau_n\}$ of n sporadic tasks. We assume that (i) time is discrete and (ii) the times when tasks are requested, are not known a priori. Any sporadic task τ_i is defined by (C_i, T_i, D_i, J_i) with:
$\star$ C_i, the maximum execution duration of the task.
$\star$ T_i, the minimum interarrival time between two requests of task τ_i, T_i is abusively called the period of task τ_i.
$\star$ D_i, the relative deadline of task τ_i. A task τ_i whose activation is requested at time t has $t + D_i$ for absolute deadline, (i.e. it must complete before time $t + D_i$).
$\star$ J_i, the maximum release jitter.

$\star$ The processor utilization factor, denoted $U = \sum_{i=1}^{n} C_i/T_i$ is the fraction of processor time spent in tasks execution. An idle time t is defined on a processor as a time such that there are no tasks whose activation has been requested

before time t, pending at time t. A busy period is defined as a time interval $[a, b)$ such that there is no idle time in (a, b) and such that both a and b are idle times.
$\star$ We define the following sets: $hp(i) = \{\tau_j, j \;/\!\!= i, priority(\tau_j) \geq priority(\tau_i)\}$ and $\overline{hp}(i) = \{\tau_j, priority(\tau_j) < priority(\tau_i)\}$. A level-i busy period is a period of activity of the processor where only tasks $\tau_j \in hp(i) \bigcup \{\tau_i\}$ are executed.

We now show how to compute the worst case response times of any sporadic task scheduled NP-FP/HPF. The notion of level-i busy period introduced by [10] for preemptive FP/HPF scheduling is extended. In a non preemptive context, a task τ_i can be delayed by a task τ_j with a lower priority having started its execution before τ_i's release. This priority inversion, called non-preemptive effect, must be accounted for.

• Feasibility and worst case response time computation

Lemma 1. *A necessary condition for the feasibility of any task set is $U \leq 1$.*

Lemma 2. *The worst case response time of any task τ_i defined by (C_i, T_i, D_i, J_i), scheduled according to NP-FP/HPF is obtained in a level-i busy period such that (i) all the tasks $\tau_j \in hp(i) \bigcup \{\tau_i\}$ are periodic with a release jitter equal to J_j and their first occurrence is generated at time $-J_j$, and (ii) one task $\tau_k \in \overline{hp}(i)$ whose duration is maximum (if any) is released at time $t = -1$.*

Proof. See [11].

Theorem 1. *Let $\tau = \{\tau_1, ..., \tau_n\}$ be a sporadic task set scheduled according to NP-FP/HPF. The worst case response time of any task τ_i is given by:*

$$r_i = max_{q=0,\ldots Q}\{w_{i,q} + C_i - qT_i + J_i\} \quad (Eq.1)$$

$$\textit{where } w_{i,q} = qC_i + \sum_{\tau_j \in hp(i)} \left(1 + \lfloor \tfrac{w_{i,q}+J_j}{T_j} \rfloor\right) C_j + max_{k \in \overline{hp}(i)}(C_k - 1) \quad (Eq.2)$$

and Q is the smallest value such that $w_{i,Q} + C_i \leq (Q + 1)T_i - J_i$.

Proof. See [11]. Notice that if $\overline{hp}(i) = \emptyset$, $max_{k \in \overline{hp}(i)}(C_k - 1) = 0$.

3.2 The Scheduling Problem

Before defining the scheduling problem, we first introduce some notations concerning the loop and disk parameters.

• Loop parameters

We consider a loop L. Let N_D be the number of disks in loop L. Let t_{fab} be the delay introduced by the fabric. As already said, this delay is assumed to be constant in a simplifying purpose. N_{dev} is the number of devices connected to a loop. $t_{through}$ is the latency introduced by each device connected to the loop. t_{prop} is the propagation delay on the loop and l_{loop} is the loop latency. The loop latency can be evaluated as follows: $l_{loop} = N_{dev} \cdot t_{through} + t_{prop}$.

Let Th_{loop} be the throughput of the arbitrated loop.
Let t_{ARB}, t_{RDY}, t_{OPN}, and t_{CLS} be the transmission times for respectively an ARB, an R_RDY, an OPN, and a CLS frame.
We now consider a device winning the loop arbitration. When this device asks for the loop arbitration by sending an ARB, it must wait for the receipt of its ARB. Hence a time $t_{ARB} + l_{loop}$.
After having won the loop arbitration, this device starts the communication by sending the OPN, waits for the receipt of a R_RDY and finally sends the frames to be transferred. Hence a time $t_{OPN} + t_{RDY} + l_{loop} + data/Th_{loop}$, where $data$ is the size in bits of the data to be transferred. After the transmission of the last frame, it finishes the communication by sending a CLS and waits for the receipt of a CLS sent by its corresponding device. Hence a time $2t_{CLS} + l_{loop}$.
Let $t_{loopctrl}$ denote the time needed to start and finish a communication on the loop. We then have $t_{loopctrl} = t_{ARB} + t_{OPN} + t_{RDY} + 2t_{CLS} + 3l_{loop}$.

- **Disk parameters**

Let s_{disk} be the seek time of the disk and l_{disk} be the rotational latency.
Let N_C be the maximum number of clients processed by any disk of loop L. We assume that m blocks of size B are retrieved in a single disk request.

- **The scheduling problem**

We want to determine the feasibility conditions associated with the scheduling on disks and on the arbitrated loop. We assume that each disk connected to the loop serves N_C clients and the loop connects N_D disks.
The worst case occurs when all the accepted clients want to retrieve a video content coded at the highest throughput Th_{video}. Let T denote the period of block transmission of a video content coded at the highest throughput. We have $T = B/Th_{video}$. With a period mT, the server generates requests asking each disk D to retrieve m blocks of size B for each of the N_C clients served by D. The disk solicited for a client in a period mT changes every mT.
We assume that the server has a buffer of β blocks of size B per accepted video stream. As soon as m blocks are transmitted to the client, the server asks the disks to retrieve the m following blocks. In order to avoid client starvation, these blocks must be received by the server before the $\beta - m$ remaining blocks in the buffer be transmitted to the client. Hence a deadline equal to $(\beta - m)T$ with the condition $\beta - m \geq 1$.
We assume that the memory available on the disk is sufficient to store the data retrieved from the disk before transmission on the loop. For each client it has to serve, a disk positions the head, reads m blocks and copies them in its memory. It then asks for the loop arbitration to transfer them toward the server. After winning the arbitration, it starts a communication with the server, transfers the requested blocks and then finishes the communication.
On the loop, the server has the highest priority. Each time the server wants to transmit a request, it asks for the loop arbitration. After winning the arbitration, it starts a communication with a disk, transfers the requests to this disk, finishes the communication and then releases the loop. The server proceeds in the same way for each request. The server sends one request per client served by a disk.

The resulting feasibility conditions are expressed in the following, assuming that all the disks serve the same number of clients, this number being the maximum possible for a disk in a given loop configuration.

3.3 Feasibility Conditions

At each period mT, the server sends one request per client served by each disk in the loop L and each disk has to serve N_C clients in this period. Each disk is assumed to use the FIFO scheduling policy. We consider two different feasibility conditions. The first one concerns the condition imposed on the utilization factor of each considered resource (disk and loop). The second one concerns the end-to-end response time for the retrieval of m data blocks for a client of the multimedia system. This time is the time elapsed between the server request time and the reception time by the server of the requested data. This end-to-end response time must meet the deadline as expressed in section 3.2.

- **Conditions on the Disk utilization factor**

We apply the results given in section 3.1 for sporadic tasks. We first express the fact that for each considered resource, the utilization factor is less than or equal to 1 (see lemma1 in section 3.1). As all the requests coming from a server are sporadic with a period of mT, this condition can be written: the workload on each resource in a period is less than or equal to the period duration.

Each disk must serve N_C clients in a period mT. The service duration of a client is equal to $s_{disk} + l_{disk} + mB/Th_{disk}$. Hence the condition on the disk utilization factor can be written: $N_C(s_{disk} + l_{disk} + mB/Th_{disk}) \leq mT$. We then obtain the maximum number of clients accepted by a disk:

$$N_C \leq \frac{mT}{s_{disk}+l_{disk}+mB/Th_{disk}} \quad (Eq.3).$$

Moreover, the worst case response time of a request is obtained when the N_C requests are received simultaneously by the disk. It is equal to $X_D = N_C(s_{disk} + l_{disk} + mB/Th_{disk})$. The best response time is equal to $s_{disk} + l_{disk} + mB/Th_{disk}$.

- **Conditions on the Loop utilization factor**

On the loop, we have:

⋆ $N_C N_D$ Server tasks of duration $C_{server} = t_{loopctrl} + t_{req}$,

⋆ $N_C N_D$ Disk tasks of duration $C_{disk} = t_{loopctrl} + mB/Th_{loop}$.

All these tasks have a period mT and the Server tasks have no release jitter.

The condition on the loop utilization factor can be written: $N_C N_D(C_{disk} + C_{server}) \leq mT$. Hence the maximum number of disks accepted by a loop is given by equation 4:

$$N_D \leq \frac{mT}{N_C(t_{req}+mB/Th_{loop}+2t_{loopctrl})} \quad (Eq.4).$$

- **End-to-end response time**

We now determine the worst case response time for the response of a disk to the server. The worst case response time between a server request and the receipt by the server of the requested data can be evaluated by means of the holistic approach [14]. This response time consists of three parts X_R, X_D and X_L where:

⋆ X_R is the latest reception time of a server request by a disk,

⋆ X_D is the disk worst case retrieval time,

⋆ X_L is the additional worst case time needed to transfer the requested data to the server (including loop and fabric transfer).

X_R can be expressed as follows: $X_R = t_{fab} + N_C N_D t_{loopctrl} + t_{req} N_C N_D + mB/Th_{loop} + t_{loopctrl}$, where $t_{loopctrl} + mB/Th_{loop}$ corresponds to a non-preemptive blocking factor due to a disk having just started its transmission on the loop when the server decides to transmit the disk requests.

X_D can be expressed by considering the last client served by a disk in a period of duration mT. We have: $X_D = N_C(s_{disk} + l_{disk} + mB/Th_{disk})$ for the FIFO policy.

In the worst case, time X_L is obtained considering a disk that gains the loop arbitration after the other disks. The server and the disks have tasks of period mT. We also consider that the server generates its requests with no release jitter, and the disks receive the requests with a jitter J_{disk}. We now determine this maximum jitter J_{disk} for a disk accessing the loop:

⋆ In the worst case, the demand to transmit on the loop the server request for a client is processed after a delay$X_R = t_{fab} + N_C N_D C_{server} + C_{disk}$. The first term is due to the fabric, the second one means that this demand is the last one among the $N_C N_D$ demands to be served. The third term corresponds to the non-preemptive effect: when the server asks for the loop transmission, a disk has just started to transmit its m blocks. According to the FIFO scheduling, this request is served after a maximum delay of $X_D = N_C(s_{disk}+l_{disk}+mB/Th_{disk})$. Hence the read blocks are ready to be transmitted on the loop after a maximum delay of $X_R + X_D$.

⋆ In the best case, the demand to transmit on the loop the server request for a client is processed after a delay $t_{fab} + C_{server}$. The disk has read the requested blocks after a delay $s_{disk} + l_{disk} + mB/Th_{disk}$. Hence the read blocks are ready to be transmitted on the loop after a delay of $t_{fab} + C_{server} + s_{disk} + l_{disk} + mB/Th_{disk}$.

⋆ The disk jitter J_{disk} is obtained by the difference between the worst case and the best case: $J_{disk} = (N_C N_D - 1)C_{server} + C_{disk} + (N_C - 1)(s_{disk} + l_{disk} + mB/Th_{disk})$.
We can apply theorem1 to the considered disk, to compute $w_{disk,q}$ the latest starting time of the q^{th} iteration of a Disk task:
$w_{disk,q} = q \cdot C_{disk} + N_C N_D(1 + \lfloor w_{disk,q}/(mT) \rfloor)C_{server} + (N_C - 1)(1 + q)C_{disk} + (N_D - 1)N_C(1 + \lfloor (w_{disk,q} + J_{disk})/(mT) \rfloor)C_{disk}$.

⋆ In the formula giving $w_{disk,q}$ the first term stands for the workload induced by the considered client served by this disk in the q previous iterations and the second term represents the workload induced by the server. The third term stands for the workload induced by the $(N_C - 1)$ other clients of the considered disk and the fourth term accounts for the workload induced by the N_C clients of the $N_D - 1$ other disks.

$\star$ The stop condition is given by Q the smallest integer value such that $w_{disk,Q} + C_{disk} \leq (Q+1)mT - J_{disk}$.

We then get: $X_L = t_{fab} + max_{q=0..Q}(w_{disk,q} - qmT) + C_{disk}$. According to the holistic approach, the end-to-end response time can be upper bounded by $X_R + X_D + X_L$. We now express the constraint related to the end-to-end deadline $(\beta - m)T$, leading to $X_R + X_D + X_L \leq (\beta - m)T$. We finally get:

$$N_C N_D(t_{loopctrl} + t_{req}) + 2(mB/Th_{loop} + t_{loopctrl} + max_{q=0..Q}(w_{disk,q} - qmT) + N_C(s_{disk} + l_{disk} + mB/Th_{disk}) + 2t_{fab} \leq (\beta - m)T \quad (Eq.5).$$

3.4 Model Validation and Performance Results

In this section, we compare the performance results obtained in our analysis with simulation results already published in [12] and [13]. These comparisons are made for different configurations. After this validation, we study the influence of different parameters on the performance of the VoD system. The maximum coding throughput considered for the video contents stored in the multimedia system is equal to 3 Mbps. In all the experiments, the size B of the block is equal 64 kBytes, leading to $T = 174.7ms$. We consider different values of m. In all graphs, we represent the total useful throughput as a function of the number of disks in the loop. The number of clients accepted by the VoD system is equal to the total useful throughput divided by the maximum coding throughput of video contents. In our experiments, we use three types of disks, whose parameters are given in the following table. Disks D_1 and D_2 are Seagate disks used in [12]. Disk D_3 is an IBM Ultrastar XP disk used in [13].

	D_1	D_2	D_3
seek time (ms)	10.5	8.34	8.5
rotational latency (ms)	5.5	4.15	4.17
sustained throughput (Mbps)	33.6	58.8	52.04

• Model validation

We represent the total useful throughput obtained on the loop considering different disk numbers, different disk parameters and different sizes of block. We compare our results with the results of [12] in figure 2a and the results of [13] on figure 2b. On figure 2a $m = 2$, leading to the retrieval of 128 kBytes for every read access. On figure 2b, $m = 1$. The results are very close and show that our model is valid for different configurations.

• Influence of the deadline

In this experiment illustrated by figure 3, $m = 2$, and the deadline is equal to $2T$, $4T$, $6T$ and $8T$. As long as the number of clients meets Eq3 and the number of disks meets Eq4, an increase of the deadline makes easier Eq5 and therefore improves the maximum number of accepted flows. For a small number of disks (less than 13 in figure 3), the deadline influence is not significative. Indeed, Eq5

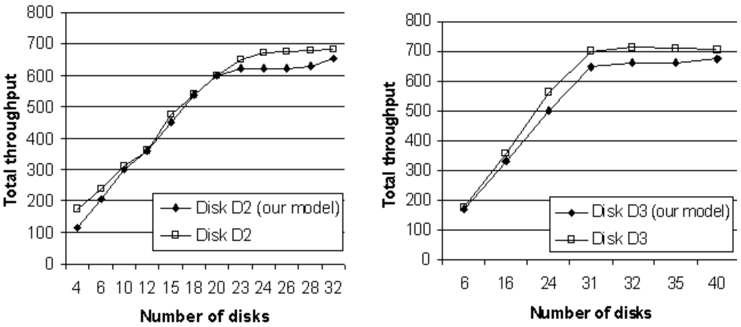


Fig. 2. Model validation a) for disk D_2 and b) for disk D_3

that is the only equation accounting for the deadline, is not the limiting one. For a higher number of disks, a deadline increase improves the performances. However, a deadline higher than $6T$ does not improve significatively the number of accepted clients.

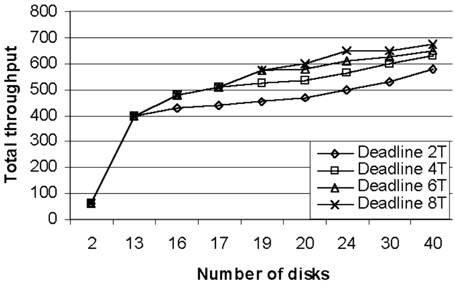


Fig. 3. Influence of the deadline

Moreover, there is a limitation imposed on the server buffer size which determines the latency for a play command. Indeed, the server buffer reserved for a client must be filled before the video content visualization starts. Hence, the optimal size is determined as a trade-off between the maximum latency acceptable by a client and the maximum number of flows accepted by a VoD system.

- **Influence of disk performance**

In this experiment illustrated by figure 4a, $m = 2$, the deadline is equal to $6T$. This experiment shows the influence of disk performance on the number of accepted clients. In Eq3 and Eq5, the best performance of the VoD system is obtained for disks providing the smallest value of $s_{disk} + l_{disk} + mB/Th_{disk}$.

This is achieved for disk D_3. A high performance disk is more interesting when the number of disks is less than 30. Over this threshold, the loop becomes the limiting factor.

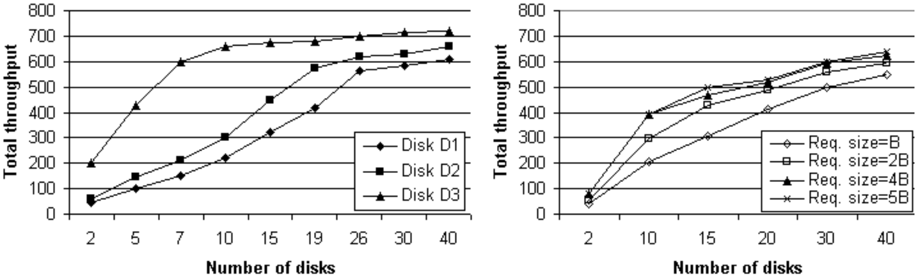


Fig. 4. Influence of a) the disk performance and b) the number of read blocks

- **Influence of the number of read blocks**

In this experiment illustrated by figure 4b, we consider different values of m ($m =$ 1, 2, 4 and 5), the deadline is equal to mT. A value of m higher than 4 does not significatively improve the number of accepted flows and the useful throughput. With $m = 4$, we have a deadline equal to 698.8 ms which is acceptable for the response time of the play/start command. Nevertheless, a high value of m is not necessarily suitable as it influences the server buffer size and the disk buffer size.

Conclusion

In this paper, we have proposed to use Fibre Channel technology in multimedia systems offering Video on Demand services and ensuring a deterministic QoS. A SAN architecture offering a good scalability has been defined. We have shown how to dimension the arbitrated loops of the SAN. This dimensioning is established from a uniprocessor real-time scheduling analysis applied to two crucial resources: magnetic disks and FC arbitrated loop. Our analysis has been validated by comparing our results with previously published simulation results. We have then computed the maximum number of clients acceptable by a loop, depending on the disk (number and performance). The optimal number of disks connected to a loop has been determined. Our analysis can be used as a dimensioning tool for a VoD system. More precisely, the results can be used to implement an admission control for new VoD clients.

References

1. A. F. Benner, "Fibre Channel for SANs", McGraw-Hill, 2001.
2. S. Wilson, "Managing a Fibre Channel Storage Area Network", http://www.sansolutions.com/SNMWG/Downloads/SAN_White_Paper-V.05.pdf
3. J. Gafsi, "Design and performance of large scale video servers", Ph. D Thesis, ENST Paris, France, Nov. 1999.
4. S.A. Barnett, G.J. Anido, P. Beadle, "Predictive call admission control for a disk array based video server", Multimedia Computing and Networking, San Jose, California, Feb. 1997.
5. B. Ozden et al., "Disk striping in video server environments", IEEE Conf. on Multimedia Systems, Hiroshima, Japan, June 1996.
6. D. R. Kenschammaana-Hosekote, J. Srivastava, "I/O scheduling for digital continuous multimedia", Multimedia Systems 5, pp. 213-237, 1997.
7. S. Sengodan, V. O.K. Li, "A quasi-static retrieval scheme for interactive VOD servers", Computer Communications, 20, pp. 1031-1041, 1997.
8. H. M. Vin, A. Goyal, P. Goyal, "Algorithms for designing multimedia servers", Computer Communications, 18(3), pp. 192-203, March 1995.
9. R. Wijayaratne, A. L. N. Reddy, "Integrated QoS management for disk I/O", IEEE Int. Conference on Multimedia Computing and Systems, pp. 487-492, Florence, Italy, June 1999.
10. J.P. Lehoczky, "Fixed priority scheduling of periodic task sets with arbitrary deadlines", Proc. of 11th IEEE Real-Time Systems Symposium, Lake Buena Vista, FL, USA, pp. 201-209, Dec. 1990.
11. L. George, N. Rivierre, M. Spuri, "Preemptive and non-preemptive real-time uniprocessor scheduling", INRIA Rocquencourt, RR 2966, France, Sept. 1996.
12. S. Chen, M. Thapar, "Fibre channel storage interface for video-on-demand servers", Proc. of Multimedia Computing and Networking, San Jose, CA, Jan. 1996.
13. D.H.C. Du, J. Hsieh, T. Chang, Y. Wang and S. Shim, "Performance study of serial storage architecture (SSA) and fibre channel arbitrated loop (FC-AL)", Computer Science Dept., Univ. of Minnesota, TR96-074, 1996.
14. K.Tindell, J. Clark "Holistic schedulability analysis for distributed hard real-time systems", Microprocessors and Microprogramming 40, 1994.

On the Resource Efficiency of Explicit Congestion Notification

Kostas Pentikousis and Hussein Badr

Department of Computer Science, Stony Brook University,
Stony Brook, NY 11794-4400, USA
{Kostas, Badr}@CS.StonyBrook.Edu

Abstract. Explicit Congestion Notification (ECN) for IP networks has received considerable attention in recent years, and has been shown to improve TCP goodput. Previous studies have centered on scenarios in which TCP with ECN (TCP/ECN) traffic competes with ECN-unaware traffic. This paper presents case studies in which moderately short flows are all ECN-capable, and compare them with the corresponding cases where the flows are ECN-unaware, running over drop tail and Random Early Detection routers. Using transmission overhead metrics, we show that TCP/ECN uses network resources more efficiently. We also consider the case of battery-operated devices and show that TCP/ECN is more power conserving than standard TCP. An unexpected outcome of our experiments is that goodput does not improve in an all-TCP/ECN environment.

1 Introduction

In the Transmission Control Protocol (TCP), senders use packet drops as a decisive indication of congestion in the network. Upon detection of segment loss, the TCP sender slows down its sending rate in an attempt to prevent congestion collapse [1]. Dropping packets to indicate congestion is a wasteful use of network resources. Furthermore, by equating segment loss with network congestion, TCP fails to perform well in networks where random errors are introduced by faulty hardware, or because the transmission media are less reliable than copper cables or fiber. Consequently, Explicit Congestion Notification (ECN) for IP networks [12] was proposed in order to provide TCP senders with clear indication of prevailing congestion in the network. ECN has received a considerable amount of attention in recent years, because it can help to increase network utilization and improve TCP performance (under traditional and emerging network technologies) [2][6][13][14], and because it might enable the development of quality of service differentiation schemes in IP networks [8].

This paper focuses on the resource efficiency of ECN in IP networks. Although one may intuitively surmise that there should be gains with ECN since packet drops are largely avoided, this has not been formally investigated, let alone quantified. We show that in an IP network where all hosts support ECN, network resources are used more efficiently with respect to criteria not considered in previous studies, which tend to focus primarily on throughput/goodput; for example, transmission overhead from

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 588-599, 2002.

dropped and duplicated packets may be reduced by as much as 70%. Conversely, a significant, unexpected conclusion from our work is that average TCP throughput/goodput is not improved with respect to relatively short flows.

2 Related Work

ECN is integrally dependent on an active queue management (AQM) mechanism [4], which determines when packets should be marked [12]. In practice, in most studies published to date, the AQM underlying ECN has been Random Early Detection (RED) [7]. Slim and Ahmed [13], for example, use a Linux-based test bed network to study the performance advantage of TCP with ECN (TCP/ECN) in terms of throughput for both bulk and transactional transfers. Key findings include: increased relative advantage for TCP/ECN over standard TCP when congestion levels increase; limited number of retransmission when ECN is employed; and smaller amount of time spent in error recovery.

Zhang and Qiu [14] evaluate TCP performance under RED and drop tail (DT), examining a variety of scenarios, including long and short transfers, different round trip times (RTTs), a mix of TCP and UDP traffic, and two-way traffic. They also use TCP goodput as their metric of ECN efficiency. Though focusing mainly on RED with packet drops, they find that when *n* TCP/ECN senders compete with *n* ECN-unaware TCP senders, the former achieve up to 30% better goodput than the latter.

More recently, Athuraliya *et al.* [2] present an evaluation of TCP/ECN as part of a study of their proposed Random Exponential Marking (REM) AQM. They focus mainly on illustrating that REM performs better than RED, and that ECN can help to further reduce losses in the network while preserving high goodput. They consider the case where infrequent random drops are happening in the network (thereby roughly simulating errors in a wireless environment), and conclude that ECN can help to increase TCP's performance in hybrid wired/wireless networks [11].

A common, determinant characteristic of the published literature known to us is that it examines scenarios in which TCP/ECN flows compete with ECN-unaware ones. The literature makes clear that, under these circumstances, the TCP/ECN sender is virtually assured better goodput. All other things being equal, a TCP/ECN sender possesses more information about network conditions and avoids decreasing its congestion window more than once per window of data [12].

Another point worth highlighting is that most studies use "background", non-ECN traffic in order to maintain a level of congestion that ensures routers operate in the "RED region": that is, with average queue sizes always within the range of values for which random early marking is in effect. This permits ECN to display its best potential with respect to the ECN-capable flows present. However, it is known that tuning RED parameters so that routers in real networks, experiencing a variety of traffic mixes, are always operating in the RED region is difficult [5][9].

This paper presents case studies using simulation in which there is no background traffic. Traffic in our studies is ECN-capable, and is not constructed to ensure any particular, *a priori* relationship with respect to the RED parameters in effect. We also focus on moderately short (100 KB) transfers, typical of, for example, web browsing. To the extent that a file transfer may terminate before the full potential of the feedback marking mechanism takes effect, shorter downloads are less favorable to ECN than longer ones. Our aim is to evaluate TCP/ECN performance under circumstances that are not necessarily inclined in its favor. The ECN cases are then compared to the corresponding cases where the traffic is ECN-unaware and running under, on the one hand, DT and, on the other, RED.

3 Experimental Configuration

Our study of TCP/ECN performance was conducted with the widely used ns-2 simulator [10]. Fig. 1 depicts the network topology, representing a situation in which multiple clients access a web server. Our network topology is similar to that used in many protocol evaluations, including the ones discussed above.

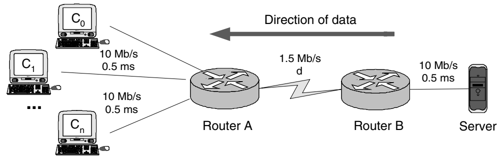


Fig. 1. Simulation topology

A fixed number of clients (receivers) *n* are each connected to router A through a 10 Mb/s link with a propagation delay of 0.5 ms. A server is connected to router B, also with a 10 Mb/s link and 0.5 ms propagation delay. Router B is connected to router A by a 1.5 Mb/s link (the “bottleneck link”). We experimented with a variety of propagation delays *d* for this link, ranging from 2 to 20 ms.

The server employs TCP Reno with a maximum segment size (MSS) of 1460 bytes. The clients use the delayed acknowledgements algorithm and advertise an initial receiver window of 64 KB [1]. Data transfers are unidirectional. Starting at time 0, each client initiates a 100 KB file transfer. When a client receives its file, it immediately initiates a new transfer for another 100 KB file. The three-way TCP connection-establishment handshake is simulated for every file transfer. An individual experiment terminates when the *n* parallel sets, each of 11 serial transfers, successfully complete.

We experiment with different values for the size of the router B output queue at the bottleneck link (*i.e.* in the direction of the data flow), using DT, and RED with and without ECN support. All other buffer queues in the network are DT, but in fact are adequately provisioned to ensure that no drops occur. This is in order to isolate the impact on performance induced by the router B queue management mechanism.

3.1 Simulation Methodology

As already mentioned, an individual experiment consists of n parallel sets, each composed of 11 serial file transfers, for a total of $11n$ transfers. Although the n sets start simultaneously, "synchronization" between them is quickly lost: first, because the SYNs of the initial n connection establishment phases are serialized as they pass through the routers on their way to the server; and secondly because the $11n$ transfers undergo different experiences at the bottleneck router, yielding a rich mixture and variety of TCP congestion and error-recovery responses. This, for example, is illustrated in Fig. 2: the distribution of ECN marks across the possible sequence numbers is close to uniform for the vast majority of experiments. At any given instant of an experiment (at least until the first of the n parallel sets completes its 11 transfers) there are n simultaneous, ongoing downloads at different phases of progress and TCP induced dynamics. As such, the $11n$ transfers provide a rich sampling of possible outcomes. The results reported here, mostly averages for a single 100-KB file download are based on aggregation over this sample of size $11n$.

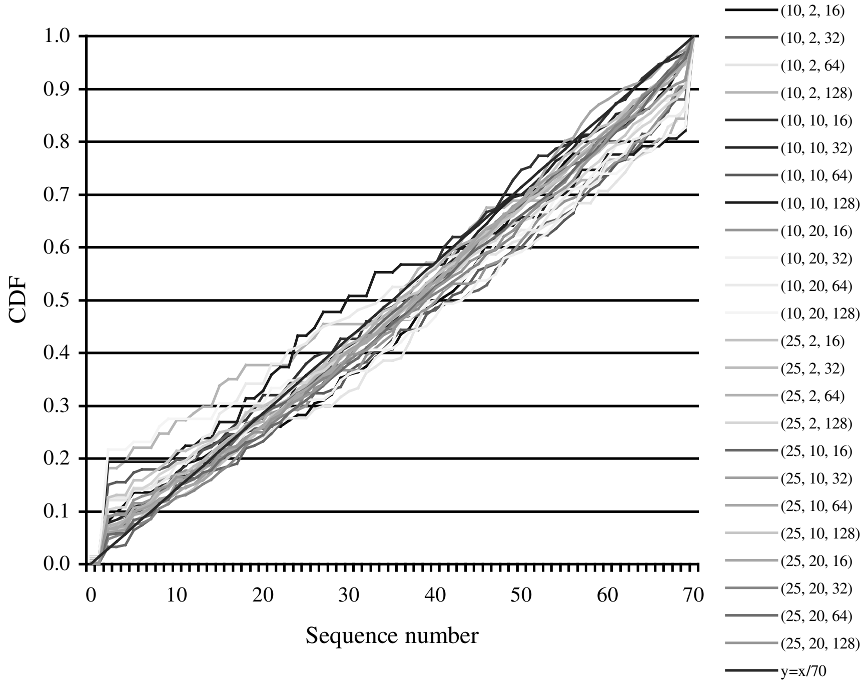


Fig. 2. Cumulative distribution function (*CDF*) of ECN marks across a number of experiments. Experiment configuration parameters are given as (number of clients, d in ms, QL at router B in packets). See also TABLE 1

It should be noted that very little in the simulation is driven by random numbers: ns-2 runs actual TCP code, and packet transfer times are deterministic for a given link, once its transmission rate and propagation delay are defined. The only aspect of the simulation that receives generated random variate input is the RED marking at the router B buffer. Because of this, file transfer averages of the kind we report on would not vary significantly across independent replications of the experiment, and the results we present, based on a single replication, are quite representative (representative, that is, for the particular case study network configuration we are simulating). In the few cases where we implemented an independent replications procedure, samples based on 12 replications yielded small variance. The 98% level confidence intervals for single-file transfer times, for example, were mostly within ±5% of the sample means.

Of course, a single replication's 11*n* sample values are not mutually independent since they result from the interaction of *n* parallel sets of serial TCP transfers competing for network resources. They are not even identically distributed since, as each client completes its 11 file transfers and "shuts down", the remaining clients experience less competition for network resources. As such, we have not attempted to formally wrap the (single-run) averages reported in confidence intervals.

3.2 Parameter Setting

Our experiments are primarily aimed at comparing the performance of standard TCP (over DT and RED) with that of TCP/ECN. The number of clients ranges from 5 to 25, and the buffer size (QL) at router B from 8 to 128 packets. RED parameters [7] were set as follows: the minimum ($\min_{th}$) and maximum ($\max_{th}$) thresholds varied depending on QL, see Table 1; 10% for the maximum dropping probability, $\max_p$, for all tests presented in this paper; and a weighting factor, w_q, used to calculate the average queue size, fixed at 0.002 for all tests. RED was always configured with the "gentle" option [10].

Table 1. RED parameter settings

QL	16	32	64	128
$\min_{th}$	5	10	20	40
$\max_{th}$	15	30	60	120

4 Results and Discussion

Our primary interest in this paper is to compare TCP when complemented by a "pure" marking mechanism to the more usual situation of "standard" TCP with a dropping mechanism. The former is represented by TCP/ECN, and the latter by TCP over DT. TCP over RED, as such, is ancillary to our focus because it still uses dropping as a congestion notification mechanism. ECN is dependent on an AQM mechanism, in this case RED's. As such, results from TCP over RED provide a "control" group which enable us to apportion the gains achieved by TCP/ECN between, on the one hand, its AQM scheme and, on the other, the marking mechanism per se. Due to space limitations, however, we only present results for 10 and 25 clients. Results with fewer clients are different to the extent that they entail lower levels of congestion, and will be briefly discussed below. Results with more clients are similar with the ones we present in this paper, if QL is increased. It should be noted, however, that with 10 clients the router B buffer operated in the RED region most, but not all, of the time, and with 25 clients virtually all of the time.

4.1 Transfer Duration

In contrast to previously published studies [2][6][13][14], our experiments produced results that consistently show no significant improvement in TCP goodput, across different scenarios, when ECN is employed. In fact, in some cases the average transfer times to complete the 100 KB download are somewhat larger (and hence the average goodput smaller) for TCP/ECN. For example, Fig. 3 presents the average transfer times (and the standard deviations) for 10 and 25 clients, with propagation delay *d* along the bottleneck equal to 20 ms, for four varied QLs.

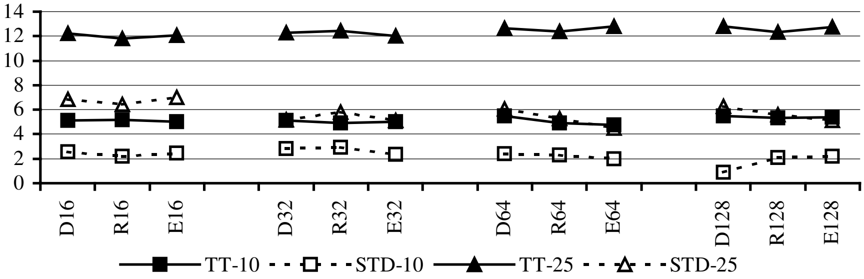


Fig. 3. Average transfer duration (*TT*) and standard deviation (*STD*), in seconds, with *10* and *25* clients competing (*d* = 20ms). (*D*: DT; *R*: RED; *E*: ECN)

At first sight, it seemed intriguing that TCP/ECN does not lead to improved performance. Closer examination of the experiment data revealed that TCP/ECN indeed does become aware of congestion conditions sooner than does standard TCP and, therefore, backs off in a more timely manner (see subsection 4.2, below). However, it is unable to detect improved network conditions any more efficiently than standard TCP since, in this respect, both implement the same mechanism for congestion window expansion. Earlier studies have shown that, when TCP/ECN competes with ECN-unaware traffic (*i.e.*, over RED), it has the advantage in detecting congestion and so avoids transmitting in times of congestion. This in turn helps it avoid packet drops, and thus prevents TCP's congestion window collapse, yielding increased goodput. On the other hand, when all traffic is TCP/ECN none of the senders has a competitive advantage and "the gains of one flow are the losses of another". For any given QL, TCP over DT undertakes more segment transmissions overall in order to complete the 11*n* transfers than TCP/ECN (see subsection 4.2, below). With less timely information about congestion, TCP over DT attempts some segment transmissions that TCP/ECN does not (since the latter halves its congestion window immediately upon receipt of ECNs). Some of these "risky" (because of rising congestion levels) transmissions succeed, allowing TCP to achieve on the average similar goodput with TCP/ECN.

Furthermore, TCP's highly tuned algorithms, combined with flow multiplexing, yield bottleneck link utilization in the mid- to high ninety percent range (especially when 25 clients are involved), leaving very limited spare bandwidth, which causes the average transfer times to vary hardly at all across varying QLs. Note that these QLs are sufficiently large to ensure that bottleneck link starvation does not occur.

4.2 Network and Receiver Overhead

We define network overhead as the number of excess packets that the TCP sender transmits for the specified application payload (100 KB), over and above the minimum required in an uncongested network. Network overhead is expressed as a percentage of this minimum, and includes packets that are dropped at the bottleneck router, packets corrupted in transmission, as well as duplicate (*i.e.* unnecessarily retransmitted) packets. In contrast, receiver overhead is defined as the number of packets received at the client in excess of the application payload minimum, also expressed as a percentage of the latter. Thus, receiver overhead includes duplicate and corrupted packets, but not packets dropped at router B. Both overheads are presented in Fig. 4 below for the case of 10 and 25 clients, with *d* = 20 ms.

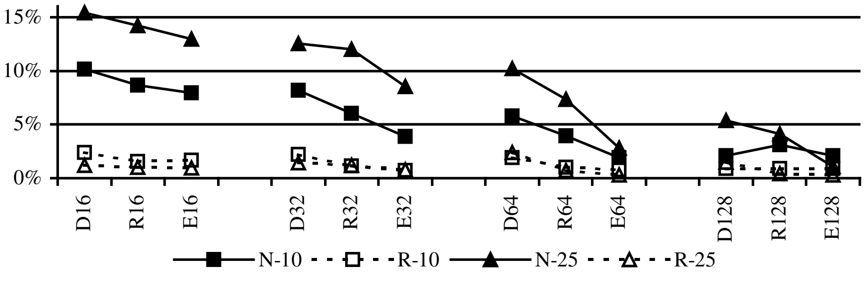


Fig. 4. Network (*N*) and receiver (*R*) overhead with *10* and *25* clients competing (*d* = 20ms). (*D*: DT; *R*: RED; *E*: ECN)

Taking network overhead first, and as can be seen from the figure, TCP/ECN induces smaller overhead than TCP does. Consider, for example, the case of QL = 64. The TCP network overhead is 10.18% over DT, and 7.32% over RED; TCP/ECN's is 2.78%. That is, network overhead is reduced by as much as 72%, yielding 7.40% fewer total transmitted packets. Also note that in this particular case, TCP/ECN absolute gains (in terms of number of packets) increase as more clients compete, while relative network overhead (in terms of percentage) remains constant. With 10 clients active, TCP over DT transmits 445 packets more packets than what is required to transfer the total application payload; if TCP/ECN is used, this figure drops to 149 packets (*i.e.*, network overhead reduction of 66%). With 25 clients, the figures are 1960 vs. 535 packets, yielding a reduction of 72%.

Clearly, not all gains are due purely to the marking mechanism: RED itself achieves some gains over DT. The only exception is for QL = 128 with 10 active clients, where router B buffer can accommodate all flows with hardly any drops under DT. There are almost 13 slots in the buffer for each active flow, each of which experiences a negligible number of congestion incidents. With RED, random segment losses are induced, yielding poorer network overhead. TCP/ECN's network overhead is not affected since packets are marked rather than dropped.

Moreover, ECN yields the same or less network overhead than DT with twice the buffer size. This resource efficiency can lead to improved services. Our results confirm that, as expected in a highly congested network, the average packet transfer time is halved when QL is halved. Although interactive flows are not dealt with in this paper, the potential gains for short, transactional requests are clear.

In general, when QL is under-provisioned RED degenerates and causes more packet drops than DT, due to occasional random dropping over and above buffer overflow [5]. On the other hand, when QL is over-provisioned the network overhead is similar for ECN and DT, and slightly better than for RED with dropping. In terms of receiver overhead TCP/ECN consistently improves on TCP over DT, if only modestly: 2% of total packets received, though this represents a relative gain of as much as 88% (Fig. 4).

Of course, when congestion levels are lower, the benefits of ECN are negligible. For example, with five clients, no background traffic, and *d* small (2 ms), the bottleneck router does not operate in the RED region, yielding rather disappointing results in terms of network overhead for TCP/ECN (not shown). Increasing *d* to 20 ms, we have TCP/ECN yielding a slight improvement in network overhead.

4.3 Power Efficiency

The resource efficiency of TCP/ECN can be of greater importance in networks other than the traditional wired Internet. In particular, resource efficiency can translate into power savings for battery-operated (mobile) hosts. Although exact communications-related power consumption depends on the hardware and software specifics of the battery-operated device, it is important to use a transport protocol that is as resource efficient as possible.

With the proliferation of wireless networks, TCP is being called on to serve as the transport protocol in hybrid wired/wireless environments. It is well established in the literature that TCP Reno does not perform well in such environments, because, amongst other reasons, TCP interprets all segment losses as congestion incidents [11]. ECN can provide more information to a TCP sender about the congestion levels and therefore allow it to make more intelligent decisions. In order to see the effect of rather infrequent random losses on the performance of TCP/ECN, and compare it with the results presented already, we repeated the experiments with the addition of a rather simple, Bernoulli trials error model on the links connecting the clients with router A, *i.e.* the clients' access subnetwork.

In any battery-operated device, three factors determine communications-related power consumption: (a) the amount of packets transmitted, (b) the amount of packets received, and (c) the amount of time spent idling. In our case study, assuming that clients are battery-operated: (a) corresponds to the ACKs sent back to the server; (b) to packets received; and (c) can be associated in a straightforward manner with the transfer time. Therefore, the metrics used in the previous section can serve as rough indicators for the energy efficiency of TCP/ECN.

4.4 Receiver Overhead with Random Errors

Receiver overhead can provide us with a metric of totally wasted energy because it measures the amount of packets that are redundant. Fig. 5 presents network and receiver overhead when 1%-probability random errors are introduced in the clients' access subnetwork, and *d* is set to 2 and 20 ms, respectively. TCP/ECN achieves smaller receiver overhead across all scenarios. For example, for 25 active clients, and QL = 128, TCP receiver overhead is 3.07% for DT, 0.87% for RED, and 0.56% for ECN. In other words, the reduction in receiver overhead for TCP/ECN is almost 82%; this translates to 2.5% fewer total packets received.

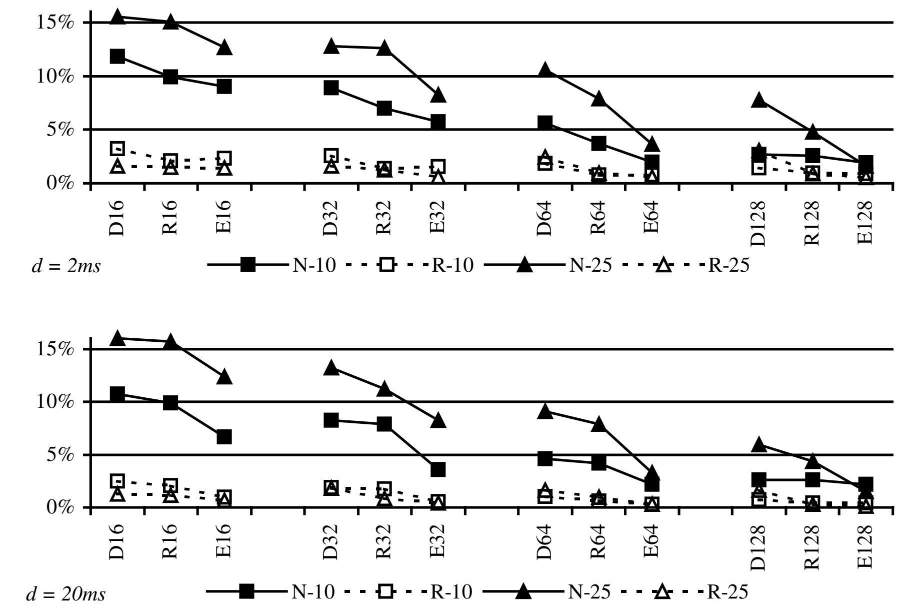


Fig. 5. Network (*N*) and receiver (*R*) overhead (*10* and *25* clients); random errors are introduced. (*D*: DT; *R*: RED; *E*: ECN)

Light random errors force TCP to behave more conservatively and so impose slightly smaller load on the network. Comparing the results presented in this section with those in subsection 4.2, we conclude that TCP/ECN still maintains an advantage, in both network and receiver overhead.

4.5 Transfer Duration with Random Errors

From an energy point of view, and all other things being equal, the smaller the amount of time needed to transfer the application payload the larger the savings. Fig. 6 illustrates the average transfer time and the standard deviation when random errors are introduced, for *d* set to 2 and 20 ms, respectively. The results are similar with the ones presented in subsection 4.1, above. There is hardly any variation in average transfer times across all scenarios when only 10 clients are active. With 25 clients active, TCP/ECN achieves slightly improved transfer times in some scenarios (*e.g.,* QL = 16), and worse ones in others (*e.g.*, QL = 64, *d* = 2ms). In general, though, TCP/ECN does not have a competitive advantage over standard TCP, and the arguments presented in subsection 4.1 apply here too. Thus, if we consider transfer times alone, TCP/ECN is no more power-efficient than TCP over DT.

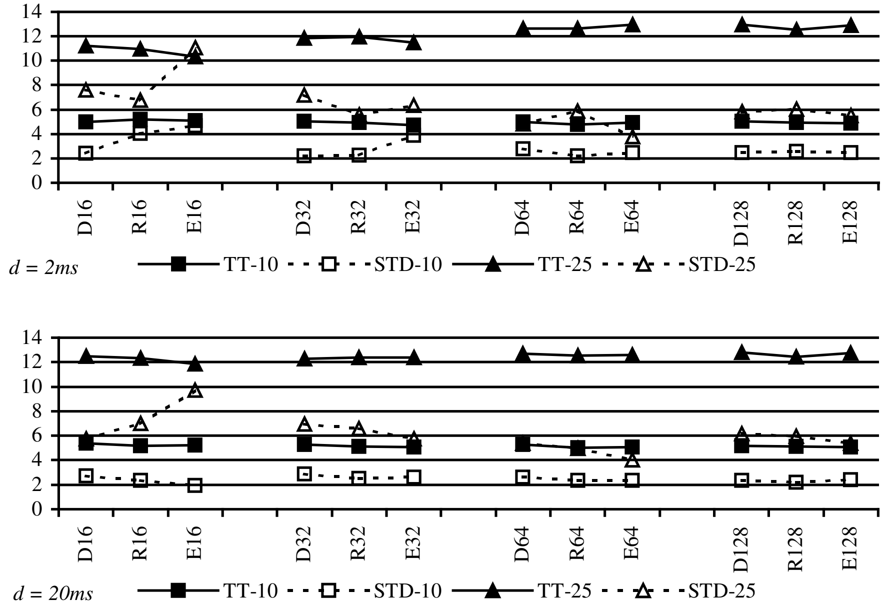


Fig. 6. Average transfer duration (*TT*) and standard deviation (*STD*), in seconds, (*10* and *25* clients); random errors are introduced. (*D*: DT; *R*: RED; *E*: ECN)

4.6 Transmission Energy Savings

The use of the delayed ACKs algorithms is very popular with most TCP implementations and has been shown to be network resource efficient. It is also deemed an efficient power-conserving mechanism. However, delayed ACKs double the amount of time that a TCP sender spends in Slow Start, which means that transfer times, especially for short files, are prolonged. In light of the results presented in this paper, we want to investigate the impact of delayed ACKs on TCP/ECN.

Our experiments assume that battery-operated clients have no data to send and always act as receivers. However, they do initiate file transfers and transmit TCP ACKs. Though ACKs are small in size (40 bytes for the IP and TCP header, plus the headers of the lower levels), they can be as costly to transmit as full sized segments. Balakrishnan *et al.* [3] note that in many asymmetric networks, such as wireless packet radio, MAC schemes introduce overhead per upstream transmission, which can make the transmitting of short packets (including TCP ACKs) as costly as transmitting MTU-sized packets. Therefore, limiting the number of ACKs that the receiver sends can translate into significant power savings.

Fig. 7 presents the average number of ACKs generated under a variety of scenarios. In all these scenarios, TCP/ECN generates fewer ACKs than TCP over DT. Moreover, TCP/ECN generates fewer ACKs than TCP over RED, except for QL = 32, d = 2ms, and 10 active clients. The maximum gains for TCP/ECN are achieved when QL = 128, d = 2 ms, and 25 clients are active: TCP over DT generates almost 54, while TCP/ECN generates approximately 44, ACKs, a reduction of 15%. Notice that part of these gains is apportioned to using RED at the bottleneck router. If QL is reduced to 64 packets, TCP/ECN clients generate (on the average) 10% fewer ACKs. Further reductions in QL translate into smaller gains for TCP/ECN.

5 Conclusion

In this paper we use simulation to study the case of moderately-short TCP file downloads with a bottleneck link in the end-to-end path. Our focus is on the resource efficiency of an all-TCP/ECN environment, as compared to all-DT and all-RED environments.

We observe, contrary to our expectations, that TCP/ECN does not achieve improved goodput. On a more positive note, we show that ECN leads to significant gains in network overhead, and modest, but measurable, gains in receiver overhead. A rough rule of thumb is that an all-TCP/ECN environment maintains the same overall performance, in terms of transmission overhead for duplicate and dropped packets, and for ACKs, as an all-DT environment with twice the buffer size at the bottleneck router.

Introducing light, random, wireless-like error conditions in the receivers' access network does not degrade ECN's resource efficiency. This suggests promising power-conserving potential for battery-operated mobile devices, especially in limiting the number of generated ACKs.

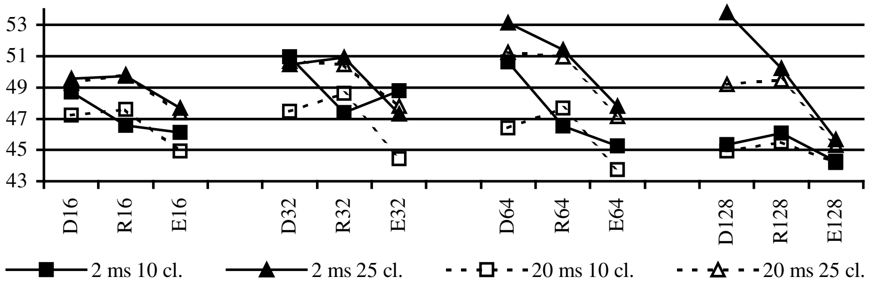


Fig. 7. Average number of ACKs sent when random errors are introduced in the clients' (*cl.*) access subnetwork. (*D*: DT; *R*: RED; *E*: ECN)

References

1. Allman, M., Paxson, V. and Stevens, W. R.: TCP Congestion Control. RFC 2581, April 1999.
2. Athuraliya, S., Li, V., Low, S., and Yin, Q.: REM: Active Queue Management. In: IEEE Network, May/June 2001.
3. Balakrishnan, H., *et al.*: TCP Performance Implications of Network Asymmetry. Internet Draft, <http://www.ietf.org/html.charters/pilc-charter.html>, (March 2002).
4. Braden, B., *et al.*: Recommendations on Queue Management and Congestion Avoidance. RFC 2309, April 1998.
5. Christiansen, M., Jeffay, K., Ott, D., and Smith, F.D.: Tuning RED for Web Traffic. In: Proceedings of ACM SIGCOMM 2000, Stockholm, Sweden, August 2000.
6. Floyd, S.: TCP and Explicit Congestion Notification. In: ACM Computer Communication Review, Vol. 24, No. 5, October 1994.
7. Floyd, S. and Jacobson, V.: Random Early Detection Gateways for Congestion Avoidance. In: ACM/IEEE Transactions on Networking, Vol. 3, No. 1, August 1993.
8. Lavens, K, Key, P., and McAuley, D.: An ECN-based end-to-end congestion-control framework: experiments and evaluation. Microsoft Research Technical Report, MSR-TR-2000-104, October 2000.
9. May, M., Bolot, J., Diot, C., and Lyles, B.: Reasons Not to Deploy RED. In: Proc. of 7th International Workshop on Quality of Service (IWQoS'99), London, UK, June 1999.
10. UCB/LBNL/VINT Network Simulator - ns (version 2). <http://www.isi.edu/nsnam/ns>, (March 2002).
11. K. Pentikousis: TCP in wired-cum-wireless environments. In: IEEE Communications Surveys, Vol. 3, No. 4, Fourth Quarter 2000.
12. Ramakrishnan, K.K., Floyd, S., and Black, D.: The Addition of Explicit Congestion Notification (ECN) to IP. RFC 3168, September 2001.
13. Slim, J. H., and Ahmed, U.: Performance Evaluation of Explicit Congestion Notification (ECN) in IP Networks. RFC 2884, July 2000.
14. Zhang, Y. and Qiu, L.: Understanding the End-to-End Impact of RED in a Heterogeneous Environment. Cornell CS Technical Report 2000-1802, July 2000.

Sender-Side TCP Modifications: An Analytical Study*

R. Lo Cigno[1], G. Procissi[2], and Mario Gerla[3]

[1] Politecnico di Torino, Dipartimento di Elettronica,
Corso Duca degli Abruzzi, 24, 10129 Torino, Italy
locigno@polito.it
[2] Università di Pisa, Dipartimento di Ingegneria della Informazione,
Via Diotisalvi 2, 56126 Pisa, Italy
g.procissi@iet.unipi.it
[3] Computer Science Department – Boelter Hall – UCLA
405 Hilgard Ave., Los Angeles, CA, 90024 – USA
gerla@cs.ucla.edu

Abstract. This paper considers a number of modifications that can be applied to the congestion control algorithm of a TCP sender without requiring the co-operation either of the network or of the receiver, analyzing their impact on the performance of the protocol. We use a theoretical approach based on the use of queueing networks for the description of the protocol dynamics and a fixed point approximation to derive the working point of the IP network. Our results show that in presence of short lived connections the impact of the transient behavior of TCP on the network performance is dominant, and major performance improvements can be obtained only if the transient behavior is improved.

1 Introduction and Motivations

Recent years have seen a large research effort focused on Internet congestion control, the TCP protocol being the pivot issue of the effort. Several papers [1,2] and RFCs [3] (just to mention some of them), were devoted to propose improved TCP versions. Other works focused on modeling either the TCP behavior [4,5,7] or the whole Internet "transfer function" [6], with the aim of gaining insight in the network dynamics, thus enhancing capabilities for designing and deploying improved protocol versions.

Several recent proposals [8,9,10] advocate the use of explicit feedback from the network, or, at least, the cooperation of the sender and the receiver (like TCP-SACK), in order to improve the congestion control. Other authors [11] claim that modifications in the congestion control algorithms should involve the TCP sender only, avoiding any necessity for co-ordination in the deployment of new TCP versions. Clearly, new TCP versions must be backward compatible and "friendly" to existent TCP implementations (TCP-NewReno in particular).

* This work was supported in part by the DARPA-MAYA (Next Generation Modeling & Simulation Tools for Global Networks) project and in part by the Italian Ministry for University and Research (MIUR) through the PLANET-IP project. The work was carried out when both Renato Lo Cigno and Gregorio Procissi were visiting U.C.L.A. during summer 2001.

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 600–611, 2002.

The focus and contribution of this paper lie in the exploration of several possible TCP modifications on the sender side that change the way TCP tries to avoid congestion and reacts to it. We do not claim that these are the only possible modifications, but they surely have a major impact on the main parameters that govern the closed-loop behavior of the TCP-sources/transpor-network system. As discussed in [6,12,13,14,15, 16] the reasons of TCP performance cannot be searched for in the protocol alone, but have to be analyzed in a context where TCP is just one piece of the overall transfer function of the closed-loop system. All the papers cited above give a deep insight in the system behavior, but the modeling technique adopted there does not include enough details to account for "apparently minor" protocol modifications and, most of all, for the transient behavior of TCP during the first slow start after opening the connection, which instead dominates the performance when short lived connections are involved.

The modeling technique we adopt, shortly described in Sect. 2, allows taking into account both the TCP transient and any kind of protocol modification. The closed-loop nature of the system is taken into account with a FPA (Fixed Point Approximation) method, that, in the case of the system under analysis, ensures the existence and the uniqueness of the solution, as demonstrated in [17].

The aim of our work is not the proposal of a specific TCP modification, but the exploration of the possible modifications and the benefits that derive from them. Moreover, this paper shows that the use of a powerful modeling technique can help in designing protocol modifications (new protocols?) without incurring the risks inherent to empirical/heuristic design.

2 The Modeling Technique

Following the approach first adopted in [7,18,19,20], in this paper we use an open multiclass queuing network (OMQN)-based description of the TCP protocol to investigate the benefits and drawbacks of possible modifications to TCP. An OMQN is a queueing network in which all queues are $M/G/\infty$. In this model, each customer (namely a TCP connection) is uniquely identified by a pair (q, c), where q represents a specific state of the protocol and c identifies the number of remaining packets to be sent before the completion of the flow. An OMQN model is capable of describing any protocol whose dynamics can be described with a Finite State Machine (FSM). The use of classes to describe the backlog of the connection allows to model short lived connections. Given the average RTT of connections and the average packet loss probability, the OMQN model defines the load offered to the IP network, as well as the throughput and the duration of connections. The model is complemented with a single- or multi-bottleneck description of the IP network loaded with the TCP connections, that, given the load, computes the average loss probability. The overall solution is obtained iterating with the FPA technique.

The OMQN modeling technique has proven to be extremely accurate [7,20], and this is the main reason that led us to choose this analytic approach for our study, instead of a more common simulation approach. Its powerful modeling paradigm and computational efficiency, associated with its proven accuracy, enable to gain the best possible insight in the protocol dynamics and in the consequences of protocol modifications with acceptable costs, while simulation studies are always limited in their scope by computational costs and difficulties in the interpretations of the results.

Any further detail about the OMQN modeling technique is superfluous in this context and we refer the interested reader to the available literature [7,18,19,20,21].

Protocol modifications are modeled in two possible ways. The first one implies modifying the protocol FSM, hence either adding or deleting queues in the OMQN. The second one does not affect the protocol states, but only its dynamics, and corresponds to the modification of the service rates of queues and the transition probabilities, rather than modifying the OMQN structure.

3 TCP Protocol Modification

The basic congestion avoidance algorithm of TCP NewReno (presently the most diffused TCP version) is AIMD (Additive Increase Multiplicative Decrease). The basic properties of AIMD algorithms, generalizing the TCP protocol, were studied in [22,23]. Several approximations are introduced for analytical tractability, the main of which is the incorrelation of the loss process with the TCP protocol, which does not allow to fully appreciate any property of the protocol that leads to a smaller loss probability.

Recently, studies like [12,15] hint to possible performance limitations of TCP, when coupled with AQM (Active Queue Management) schemes, due to oscillatory behaviors rooted in an excessive loop gain of the congestion avoidance mechanism. On the other hand, there is no doubt that explicit bandwidth feedback from the network can optimize the performance of TCP.

These observations, together with the desire to keep the congestion control algorithm in end-hosts only, without requiring the cooperation of the network, lead to propose new versions of TCP, like Vegas [2] or Westwood [11], that try to estimate the available bandwidth (or fair bandwidth share) within the network. In both schemes the major problem arises from the difficulty of correctly estimating the available bandwidth.

Finally, measures and simulations highlighted how the first slow start is often dominating the performance of the whole flow when its length is limited. This is obvious if the flow is composed of just a few packets, but, if the maximum window size is large, its effect is dominating also connections of hundreds of packets, that indeed dominate the performance of the whole Internet. To appreciate the importance of the first slow start, it must be recalled that TCP transmits $2 \times W_{fp}$ packets in slow start, where W_{fp} is the dimension of the congestion window when the first packet loss of the flow is detected. For instance, if there are no losses and the maximum window size is 50, then 100 packets are transmitted during the first slow start. With the standard Ethernet MSS, this roughly corresponds to 1.2 Mbits. Indeed what may happen, and really happens more often that one would believe, is that most of the packets of a flow are transmitted during the first slow start without losses, then, when the window is inflated enough to load the network, a burst of packets is lost, leading to a timeout.

In light of the above considerations we analyze three possible modifications of the TCP protocols, whose separate impact on performance will be discussed in Section 4:

- RFS (Restart from Fair Share), that reduces the window oscillations;
- W$_p$P (Window p-Persistant), that modifies the steady state loop gain;
- ESSE (Early Slow Start Exit), that improves the TCP transient behavior.

In the following, we formally define these modifications and describe how they can be modeled, assuming that the reader is familiar with the description of TCP through an OMQN.

3.1 RFS: Packet Drop Reaction and Bandwidth Estimation

In lack of explicit bandwidth feedback from the network, the congestion control algorithm cannot avoid probing the network capacity and including some form of window oscillations around the steady state operating point. Indeed, the fast recovery option of TCP implicitly assumes that the available bandwidth within the network is somewhere between $W_l/2 \cdot \overline{\mathrm{RTT}}$ and $W_l \cdot \overline{\mathrm{RTT}}$, where W_l is the window size when a packet loss is detected. Without discussing here the correctness of this assumption, it is clear that, if some better estimation of the available bandwidth can be obtained (and we are not concerned here on *how* it is obtained), then, after a loss event, the TCP protocol could resume the transmission from the window size corresponding to this estimation, say W_{fs} (the subscript "fs" standing for fair-share). This would reduce the amplitude and frequency of the window oscillations, and help increasing network stability and performance. Obviously, the estimation of the fair share window $\widehat{W_{\mathrm{fs}}}$ is a real value, while the window size is an integer: the rounded or truncated value can be used instead. We point out that this modification is not easily compared with the *responsiveness* of an AIMD protocol as defined in [23], since the relationship between the current window size and W_{fs} is not known, and for this reason we will avoid the use of this term.

From the protocol point of view, this modification is easy, since it only implies to assign a different value to $cwnd$ after a fast recovery. The OMQN model can take this modification into account exactly in the same way it accounts for the window thresholds distribution (see [7]), assigning to W_{fs} the value of the average window size computed in the previous iteration, i.e., by evaluating an ensemble average taken over the active protocol states and iterating the solution until convergence.

Since the available bandwidth computed by the OMQN is not subject to evaluation errors, in order to estimate the impact of bandwidth estimation errors on the performance of the resulting protocol, in the model we included an error function. The error function implementation is trivial: the transition exiting a fast recovery state is not deterministic but follows a probability distribution centered around the value of W_{fs} computed by the OMQN. Obviously, the distribution can assume only integer numbers between 2 and $W_{\max}$, where $W_{\max}$ is the maximum window size negotiated between the sender and the receiver.

To exemplify the modeling process, Fig. 1 reports the OMQN portion representing the transitions from a generic fast recovery state with window size n (queue LF_n) to the congestion avoidance states. The transition exiting under the queue corresponds to the event of loosing the re-transmitted packet and leads to a timeout (not shown in the figure for the sake of simplicity). The vector $\mathcal{E}(w)$ is a probability vector that describes the estimation errors.

In this study we consider only three simple cases of bandwidth estimation errors.

1. *Best Bandwidth Estimator (BBE).* The information on the connection's fair share is supposed to be exact. The corresponding distribution function $\mathcal{E}(w)$ is then:

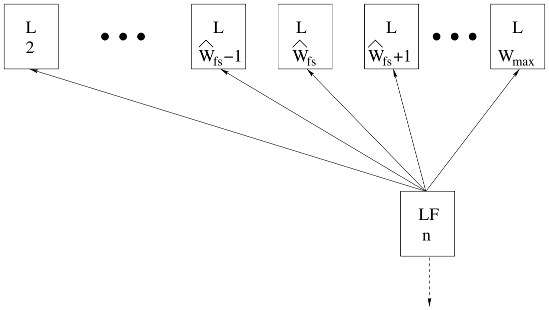


Fig. 1. OMQN description of RFS exiting a fast recovery with window size n

$$\mathcal{E}(w) = \begin{cases} 1-p & w = \lfloor \widehat{W_{\text{fs}}} \rfloor \\ p & \lfloor \widehat{W_{\text{fs}}} \rfloor + 1 \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where $p = \widehat{W_{\text{fs}}} - \lfloor \widehat{W_{\text{fs}}} \rfloor$.

2. *No Bandwidth Estimator (NBE).* This is the worst case in which he information on the connection's fair share is supposed to be unavailable, that is either there is no bandwidth estimator or the estimates are totally unreliable and cannot be used. The RFS algorithm has no 'real' clue when setting the new value of the *cwnd*. The corresponding distribution function $E(w)$ is then:

$$\mathcal{E}(w) = p \quad \forall w = 2, 3, \ldots, W_{\max} \tag{2}$$

where $p = \dfrac{1}{(W_{\max} - 1)}$.

3. *'Triangular' Bandwidth Estimator (TBE).* The information on the connection's fair share is supposed to be affected by an error function whose probability distribution is linearly decreasing, eventually reaching zero. The resulting distribution of the $\mathcal{E}$ vector is triangular, centered on W_{fs}. The distribution can be asymmetrical, emulating skewed errors:

$$\mathcal{E}(w) = \begin{cases} c_u(w - \widehat{W_{\text{fs}}} + e_u) & \widehat{W_{\text{fs}}} - e_u \le w < \widehat{W_{\text{fs}}} \\ c_o(w - \widehat{W_{\text{fs}}} - e_o) & \widehat{W_{\text{fs}}} \le w \le \widehat{W_{\text{fs}}} + e_o \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

where, e_u is the maximum admissible error underestimating the fair share, e_o is the maximum admissible error overestimating the fair share, and $\frac{p_{fs}}{2}(e_u + e_o) = 1$; c_u and c_o are the slopes of the under- and over-estimation, respectively, and are given by: $c_u = p_{fs}/e_u$ and $c_o = -p_{fs}/e_o$.

3.2 W$_p$P: Window Increase and Aggressiveness

So far we have described the modifications of the congestion control algorithm as far as its reaction to packet drops concerns. However, if the aim is the reduction of the window oscillations, the window increase during the congestion avoidance phase must be also modified. Following the theoretical results in [13], to enhance network stability, the loop gain, and hence the protocol aggressiveness, should be reduced. Moreover, we suggest that, if there is a reasonable estimation of the fair share, only a very gentle probing for extra bandwidth has to be performed in order to refrain from over-congesting the network and to assure the necessary level of friendliness with older versions of TCP.

We call this modification W$_p$P (*Window p-Persistent*), from the modeling solution in the OMQN, represented in Fig. 2, which is exactly the implementation of the p-persistent algorithm in MAC protocols: at any RTT, increase the value of $cwnd$ of 1 segment with probability p and keep the window size unchanged with probability $1 - p$.

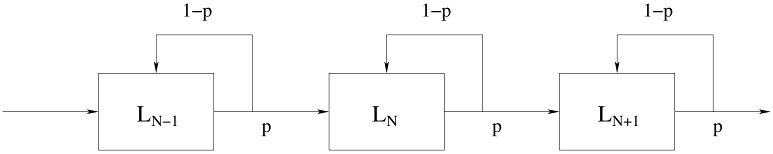


Fig. 2. OMQN representation of the W$_p$P protocol modification

The implementation in TCP can be deterministic, increasing the $cwnd$ by 1 segment every $1/p\overline{\mathrm{RTT}}$, or, as in the current TCP implementation, increasing $cwnd$ by $\dfrac{p}{cwnd}$ MSS every valid, non duplicated ACK.

3.3 ESSE: Reducing the Transient Overshoot

So far we have discussed only modifications to the steady state behavior of the congestion control algorithm, disregarding the role of the slow start phase. Internet traffic, however, is dominated by short to medium connections, so that the steady state of the network is indeed the superposition of the transients of many flows. In such a scenario, disregarding the slow start, and specially the first one, when the TCP threshold is not yet set, is extremely dangerous.

Since during slow start TCP doubles the window size at each RTT, and TCP reaction time cannot be any smaller than the RTT itself, when the threshold is not set there is an enormous potential for burst losses. In line of principle, the modification is trivial: estimate the available resources and exit the slow start to enter the congestion avoidance phase as soon as the window size reaches this estimation. We are perfectly aware that the estimation of the resources during the initial transient is extremely critical. For this reason, we assume that there is no reliable estimation until the window reaches the dimension of five segments after three RTTs. Errors in the bandwidth estimation are modeled as for the RFS case; however, the error distributions during the first slow start and during congestion avoidance are independent, so that we can model a larger error during the first slow start.

4 Assessment of the Modifications Performance

To discuss the impact of the modifications described so far, we consider two different scenarios, which are frequently encountered in the Internet. Both scenarios refer to wide area networks (WAN), since in local (LAN) environment, the congestion control algorithms of TCP very rarely play a major role. We only consider standard, FIFO queueing with drop tail policies, which are by far the most diffused queueing schemes in deployed routers. Drop tail buffers introduce correlation in losses, which are accounted for in our models with the techniques described in [7,20]. The presence of active queue management (AQM) policies and random losses due to link errors are left for future study, though they do not represent a major modeling problem.

The first scenario corresponds to the case where the bottleneck is represented by the peering point between two ISPs. This is often the case in web-browsing USA sites from Europe, since the Internet traffic is highly asymmetrical (more downloads *from* USA *to* Europe than vice-versa). The peering contracts between ISPs result in a bottleneck whose available bandwidth is nowhere near the installed capacity on transoceanic links. We assume, in this case, a bottleneck of 10 Mbit/s, with an average connections distance of roughly 10,000 km, resulting in an average RTT of 100 ms plus the queueing at the bottleneck, which is directly computed by the model and depends on the buffer size and bottleneck load. We will refer to this scenario as the PEERING scenario.

The second scenario corresponds to the case where the bottleneck of the connections is the access link of the institution where the connections originate/terminate. In this case, the bandwidth of the bottleneck can be extremely variable, depending upon the institution dimension and similar factors. We deem that this scenario is most interesting when the access link is fairly fast and, on average, there are many flows competing for the resources. We assume that the bottleneck is a 100 Mbit/s link. In this case, the average connection length is smaller, since there are both transoceanic connections and "European" connections[1]. We arbitrarily set the average length of the connections to 3,000 km, obtaining an average RTT of 30 ms. We call this scenario ACCESS. In both cases, the bottleneck buffer is set to 64 packets.

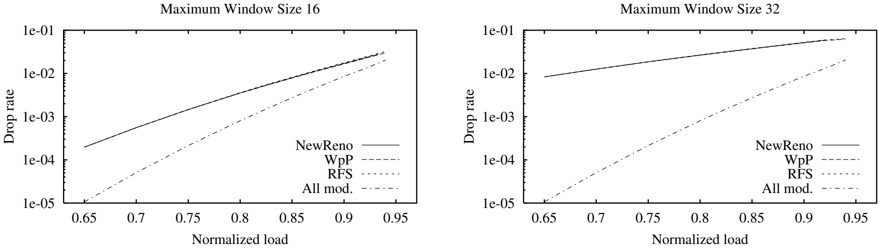


Fig. 3. Packet drop rate in the ACCESS scenario considering the W_pP, RFS modifications separately, or joint with the ESSE modification

[1] Notice that we set this hypothetical institution in Europe, but reversing the situation and having it in the USA, would not change the basic layout of the scenario

We examine first the packet drop rate in the Access scenario, considering separately the W_pP and the RFS modifications, or both of them joint with ESSE modifying also the transient behavior. Flows are all 100 packets long, but we consider two different maximum window sizes (MWS), namely 16 and 32 packets. Fig. 3 reports the packet drop rate as a function of the nominal[2] normalized traffic load. It is quite clear that the drop rate is dominated by the transient behavior: unless the ESSE modification is enabled, the loss rate is almost indistinguishable (at least in logarithmic scale) from the loss rate of NewReno connections. It is also interesting to notice that the MWS has a major impact on the loss rate, and the gain induced by the ESSE modification grows with the window size.

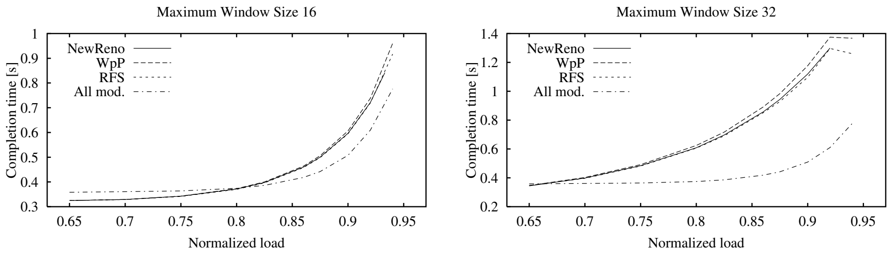


Fig. 4. Average completion times in the Access scenario considering the W_pP, RFS modifications separately, or joint with the ESSE modification

The performance gain obtained in terms of packet drop rate is striking; however, end users are more interested in the time they need to transfer the information. The two plots in Fig. 4 report the average flow completion time corresponding to the same situation of Fig. 3. It is clear that with large MWS the gain with the three joint modifications is determinant, specially at high loads (the solution of the model shows some numerical instability at very high loads, which causes the connections duration of NewReno, W_pP and RFS to decrease slightly at load 0.94).

The situation with small MWS is instead more complicated. Though not quantitatively large, the advantage at high loads is still clear when all modifications are considered together; however, at light loads, the ESSE modification seems to penalize, though only marginally, the connections. The reason is a too early exit from the slow start, that avoids connections to reach the MWS. Indeed, with small drop rates, most of the connections terminate without losses, hence the larger the reached window in slow start, the faster is the transfer. Results would probably be different considering the 90^{O} percentile of the completion time. Unfortunately, our model is not yet capable of computing completion time variances or distributions.

[2] This is the load that the bottleneck would have if there were no retransmissions, hence the actual load of the network is higher

4.1 The Influence of Bandwidth Estimation Errors

The performance of a protocol that has a reliable estimation of the available bandwidth and makes an intelligent use of this knowledge should not be a surprise; however, we are interested in analyzing the resilience of such protocols to estimation errors. The estimation errors are modeled as described in Sect. 3.1. When the errors have a triangular distribution, the maximum relative error is 25% of the average available bandwidth. We only consider all the modifications implemented together and the error estimation function is the same during slow start and congestion avoidance. The completion times of NewReno are reported for comparison.

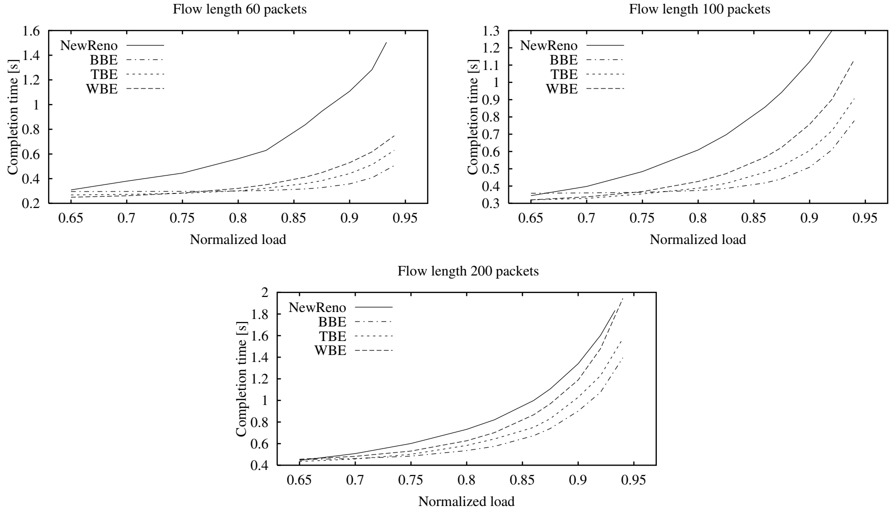


Fig. 5. Average completion times in the ACCESS scenario when all modifications are implemented, but errors impair the bandwidth estimation

We analyze here only the completion times of connections, since these also reflect the underlying loss rate.

Figs. 5 and 6 report the average completion times for connections that are 60 (top-left plot), 100 (top-right plot) and 200 (bottom plot) segments long, in the ACCESS and PEERING scenarios respectively. 'BBE' curves refer to the ideal bandwidth estimation, 'TBE' curves to the triangular error function and 'NBE' curves to the case when the estimation is uniformly distributed, i.e., drawn completely at random. The qualitative behavior is the same in all cases, regardless of the bottleneck position or capacity, or of the average connections length. The striking result is that NewReno, at high loads, always performs so poorly as to be worse than choosing the window size at random. The explanation is rooted once more in the transient behavior of the protocol, and indeed, when connections are longer, the performance gap is partially filled. NewReno, as all TCP implementations, during the first slow start grows the window until the network is congested and packets are dropped. In other words, during the initial transient, NewReno does nothing

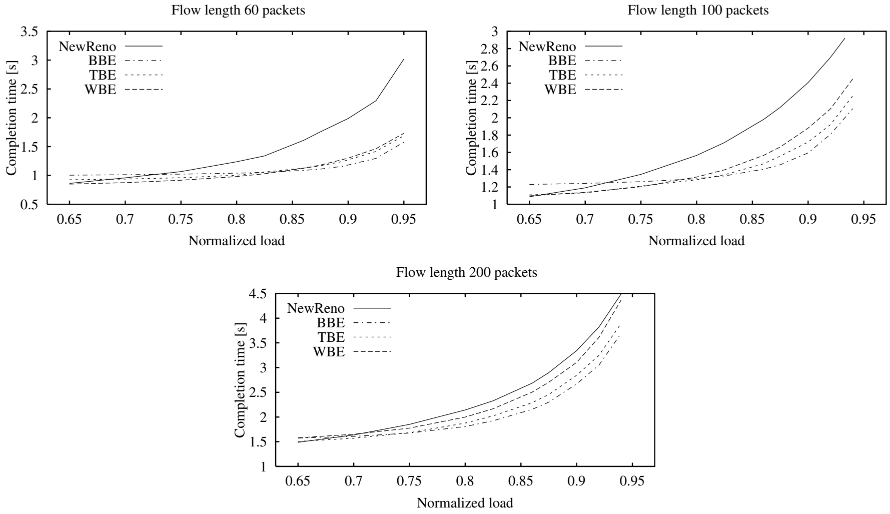


Fig. 6. Average completion times in the PEERING scenario when all modifications are implemented, but errors impair the bandwidth estimation

to estimate the available resources, but assumes they are "infinite," deterministically overloading the network; this behavior ends up in poor performance because of the higher packet loss rate, even higher than a protocol that, instead of correctly evaluating the available resources draws a uniform random variable to set the window size.

Going a little bit more into detail, we can observe that, for shorter connections and low loads, the early exit from the slow start implemented by the ESSE modification may be a little penalizing, though always marginally. Comparing the PEERING and ACCESS scenario it is clear that the larger the available resources, the higher the gain we can achieve from a correct estimation of the available resources from its use to reduce the initial transient overshoot.

5 Conclusions and Future Work

TCP is a reliable, robust and reasonably well performing protocol; however, it is far from perfect, and scores of modifications have been proposed to improve its performance. Some of them were successfully implemented and are now available in standard implementations, others did not have the same success. The best of the effort in TCP study was always devoted to heuristic modifications, and their implementation evaluated either in a simulation environment or on test beds. The work we present in this paper is a first attempt to tackle the problem from a theoretical point of view, using models that predict the impact of the modifications before implementing them and allowing a much easier evaluation of the results, since the computing effort to obtain the results is orders of magnitude smaller than that of simulations, and testbeds can only offer results for very limited settings.

We have considered three different modifications that require changes of the sender side only, thus having potentially a minor impact if deployed in the Internet. One of the major results of this work is that, in many situations, the performance of competing TCP connections is dominated by the initial transient when the threshold is not yet set. Hence, any attempt to improve TCP performance should take into account the initial transient too, while, browsing the literature, it is clear that the best part of the effort in TCP research was devoted to its steady state.

We admit that the results we have are not definitive, since we have to refine our model towards three different directions at least. First of all we have to verify the behavior of the modified TCP version when competing with NewReno implementations, in order to verify their friendliness to the existing Internet. Though not trivial, this modeling step is feasible and we hope to show also this aspect in short time. As a "side effect" of this additional modeling effort, the model will also be able to predict the performance of any mix of different length flows. Second, since the considered modifications imply the end-to-end estimation of the available bandwidth, additional studies are required on how the estimation can be carried out and what kind of errors would presumably affect this estimation. Finally, it would be extremely interesting, though it is still not clear if it is possible, to extend the modeling technique to compute not only averages over the whole flows, but also some additional information about the actual distributions of, for instance, the completion time of the connections.

Acknowledgments. The authors express their deepest appreciation to Michele Garetto, who provided the code for solving the TCP NewReno model, assistance in the code exegesis and a lot of useful discussions.

References

1. S. Floyd, "A Report on Recent Developments in TCP Congestion Control," *IEEE Communications Magazine*, April 2001
2. L. Brakmo, L. Peterson, "TCP Vegas: End to End Congestion Avoidance on a Global Internet," *IEEE Journal on Selected Areas in Communications*, 13(8), Oct. 1995.
3. M. Mathis, J. Madhavi, S. Floyd, A. Romanow, "TCP Selective Acknowledgment Options," RFC 2018, IETF, Oct. 1996.
4. J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, "Modeling TCP Throughput: A Simple Model and its Empirical Validation," *IEEE/ACM ToN*, Apr. 2000.
5. N. Cardwell, S. Savage, T. Anderson, "Modeling TCP Latency," *Infocom 2000*, Tel Aviv, Israel, March 2000.
6. V. Mishra, W. B. Gong, D. Towsley, "Fluid-based Analysis of a Network of AQM Routers Supporting TCP Flows with and application to RED," *in Proc. SIGCOMM'2000*, Aug. 28–Sept. 1 2000, Stockholm, Sweden.
7. M. Garetto, R. Lo Cigno, M. Meo, M. Ajmone Marsan, "A Detailed and Accurate Closed Queueing Network Model of Many Interacting TCP Flows," *IEEE Infocom 2001*, Anchorage, Alaska, USA, April 22–26, 2001.
8. S. Floyd, "TCP and Explicit Congestion Notification," *ACM Computer Communication Review*, V. 24 N. 5, pp. 10-23, Oct. 1994.
9. K. K. Ramakrishnan, S. Floyd, "A Proposal to add Explicit Congestion Notification (ECN) to IP," RFC 2481, IETF, Jan. 1999.

10. M. Gerla, W. Weng, R. Lo Cigno, "Bandwidth feedback control of TCP and real time sources in the Internet," *IEEE Globecom 2000*, San Francisco, CA, USA, Nov. 27 – Dec. 1 2000.
11. M. Gerla, M. Sanadidi, R. Wang, A. Zanella, C. Casetti, S. Mascolo, "TCP Westwood: Window Control Using Bandwidth Estimation," *Proc. IEEE Globecom 2001*, San Antonio, Texas, USA, Nov. 25-29, 2001.
12. S.H. Low, D.E. Lapsley, "Optimization Flow Control, I: Basic Algorithm and Convergence," *IEEE/ACM Transactions on Networking*, 7(6):861-75, Dec. 1999.
13. S.H. Low, F. Paganini, J.C. Doyle, "Internet Congestion Control," *IEEE Control Systems Magazine*, Feb. 2002.
14. S.H. Low, "A Duality Model of TCP and Queue Management Algorithms," *Proc. of ITC Specialist Seminar on IP Traffic Measurement, Modeling and Management*, September 18-20, 2000, Monterey, CA (USA).
15. C.V. Hollot, V. Mishra, W.B. Gong, D. Towsley, "A Control Theoretic Analysis of RED," *IEEE Infocom 2001*, Anchorage, Alaska, USA, April 22–26, 2001.
16. C.V. Hollot, V. Mishra, W.B. Gong, D. Towsley, "On Designing Improved Controllers for AQM Routers Supporting TCP Flows," *IEEE Infocom 2001*, Anchorage, Alaska, USA, April 22–26, 2001.
17. T. Bu, D. Towsley, "Fixed Point Approximation for TCP Behavior in an AQM Network," *ACM SIGMETRICS 2001*, June 16-20, Cambridge, MA, USA.
18. R. Lo Cigno, M. Gerla, "Modeling Window Based Congestion Control Protocols with Many Flows," *Performance Evaluation*, No. 36–37, pp. 289–306, Elsevier, Aug. 1999.
19. M. Garetto, R. Lo Cigno, M. Meo, M. Ajmone Marsan, "On the Use of Queueing Network Models to Predict the Performance of TCP Connections," *Proc. 2001 Tyrrhenian International Workshop on Digital Communications*, Taormina (CT), Italy, Sept. 17–20, 2001.
20. M. Garetto, R. Lo Cigno, M. Meo, E. Alessio, M. Ajmone Marsan, "Modeling Short-Lived TCP Connections with Open Multiclass Queuing Networks," Tech. Rep. DE/RLC/2001-4, Politecnico di Torino, June 2001.
    Available at http://www.tlc-networks.polito.it/locigno/papers/de-rlc-01-4.ps
21. M. Garetto, R. Lo Cigno, M. Meo, M. Ajmone Marsan, "Queuing Network Models for the Performance Analysis of Multibottleneck IP Networks Loaded by TCP Short Lived Connections," Tech. Rep. DE/RLC/2001-5, Politecnico di Torino, June 2001.
    Available at http://www.tlc-networks.polito.it/locigno/papers/de-rlc-01-5.ps
22. Y. Yang, M. Kim, S. Lam, "Transient Behaviors of TCP-friendly Congestion Control Protocols," *IEEE Infocom'01*, Anchorage, AK, USA April 22–26, 2001.
23. Y. Yang, S. Lam, "General AIMD Congestion Control," Tech. Rep. TR-2000-09, University of Texas at Austin, May 2000.
    Available at http://www.cs.utexas.edu/users/lam/NRL/TechReports/

Modeling a Mixed TCP Vegas and TCP Reno Scenario

Andrea De Vendictis and Andrea Baiocchi

INFOCOM Dept. - University of Roma *La Sapienza*
{devendictis,baiocchi}@infocom.uniroma1.it

Abstract. In this paper we describe and validate the analytic model of a mixed TCP Reno and TCP Vegas network scenario. There is experimental evidence that TCP Vegas overcomes the widespread TCP version, called TCP Reno, in a number of network environments. The incompatibility between TCP Vegas and TCP Reno in heterogeneous network scenarios has been also verified by means of several simulations. The model presented in this work allows to quantitatively evaluate this incompatibility, by computing the average throughput of a TCP Vegas source in presence of a concurrent TCP Reno source. This model can help us to better understand the reasons of the vulnerability of TCP Vegas in competing with TCP Reno sources.

1 Introduction

In the last years TCP congestion control has received great interest from the networking research community. A number of analytic and experimental studies have pointed out the shortcomings of TCP and they conceived changes able to cope with some TCP limitations. The recent standardized modifications are a result of these efforts (for an updated report see [1]). However, most of these modifications concerned improvements in avoiding unnecessary timeouts and fast retransmit caused by packet reordering, isolated packet loss and multiple packet loss due to temporary network congestion. Hence, the changes are intended in order to optimize the mechanisms that regulate the response to loss detection. Conversely, the mechanisms that avoid protracted network congestion (slow start and congestion avoidance) and define how TCP sources share the available bandwidth among them are quite unchanged (at least as standards).

TCP Vegas, proposed in [2], represents a valid alternative to the congestion control performed by the currently standard and most widespread version of TCP, called TCP Reno. Although it introduces new techniques into all the main mechanisms of TCP, it is fully compatible with all the standard versions of TCP, because the changes only concern the TCP sending side.

TCP Reno congestion control reacts to the network congestion only after loss detection, thus when the network congestion has already arisen. This avoids congestion collapse, since the transmission rate is reduced as soon as a packet loss occurs; however this generates an intrinsic instability of the congestion control, whose evidence is the permanent oscillation of the source transmission rate.

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 612–623, 2002.

The key idea of TCP Vegas is to prevent the packet loss (and the network congestion) by adapting the transmission rate to the value of the round trip delay experienced by the transmitted packets.

Several simulation works have verified that TCP Vegas is better than TCP Reno in terms of throughput (between 37% and 71% better than Reno), fairness, stability, packet loss probability, end-to-end delay and ability in avoiding network congestion in a very large number of network environments [2][3][4][5].

However, some works [3][4][6] also pointed out by means of simulations and experimental trials the extreme vulnerability of TCP Vegas in competing with TCP Reno sources to take the available bandwidth. The problem is that TCP Reno is intrinsically much more aggressive than TCP Vegas, because it reduces its transmission rate only after packet loss detection.

This represents the main reason why TCP Vegas cannot be widely proposed to the users as a reliable transport protocol.

Given that TCP Vegas basic approach is a sound one, there is a need for a thorough understanding of the fine tuning of its parameters and possibly for some modifications of the basic congestion control algorithm. To this end it is useful to have an accurate analytic tool for the evaluation of TCP Vegas in a mixed TCP environment.

This paper describes an analytic model to evaluate the average throughput of a TCP Vegas source interacting with a TCP Reno source in a mixed scenario.

Since the works [3][7] already modeled the competition of TCP Vegas and TCP Reno in a network environment with small bandwidth-delay product (less than the bottleneck link buffer size), we will only analyze the case of a network with large bandwidth-delay product (larger than the buffer size). This is even more interesting as the small delay-bandwidth product case, since optical networking and high speed processing devices promise very high network capacity also in the wide area network environment.

The aim is to give a quantitative measure of the vulnerability of TCP Vegas in presence of a source implementing TCP Reno and to gain insight as to what might be acted upon to reverse this outcome.

This study can represent a starting point to find mechanisms that make TCP Vegas competitive in a heterogeneous network scenario: given the distributed nature of the Internet, this is a key element to stimulate the use of TCP Vegas as a reliable transport protocol and to improve the TCP congestion control.

The rest of the paper is structured as follows. In Section 2 the TCP Reno and TCP Vegas congestion control is described. In Section 3 we present the analytic model. In Section 4 the validation of the model is shown. Finally, in Section 5 the conclusions and hints to further work.

2 TCP Vegas and TCP Reno Congestion Control

TCP Reno and TCP Vegas adopt an end-to-end closed-loop adaptive window congestion control. It is based on five fundamental mechanisms: slow start, congestion avoidance, retransmission time-out, fast retransmit and fast recovery.

TCP Reno and TCP Vegas use slow start at the beginning of the connection and whenever a packet loss is detected via timeout.

When the congestion window reaches a threshold value (called *slow start threshold*), TCP Reno and TCP Vegas enter congestion avoidance.

The maximum limit of the congestion window is advertised by the receiver to the sender during the connection.

Both the protocols can detect packet losses by means of two mechanisms: when the timeout (set when the packet is sent) expires, they reduce their congestion window to one packet size[1], then they start again in slow start[2]. Otherwise, if three duplicated acknowledgments arrive back to the sender before the timeout expiration, the protocols perform fast retransmit and fast recovery.

TCP Vegas differs from TCP Reno by the way slow start, congestion avoidance and fast retransmit are implemented.

During slow start, TCP Reno congestion window W_R increases by one packet for every incoming acknowledgment. So the congestion window has an exponential growth. During congestion avoidance TCP Reno opens its congestion window W_R linearly, because for each incoming acknowledgment it updates the congestion window by incrementing it by $1/W_R$.

In fast recovery TCP Reno halves the current congestion window and goes in congestion avoidance, without performing slow start.

The TCP Vegas congestion control is based on two parameters representing respectively the *expected* and the *actual* rate, calculated as follows:

$$Expected = \frac{W_V}{BaseRTT} \qquad Actual = \frac{FlightSize}{RTT}$$

where W_V is the value of the current congestion window, $BaseRTT$ is the minimum round trip time experienced by the connection, $FlightSize$ is the number of packets currently not acknowledged yet, RTT is the round trip time experienced by the considered packet.

For each incoming acknowledgment, TCP Vegas computes the normalized difference $Diff$ between $Expected$ and $Actual$:

$$Diff = (Expected - Actual) \cdot BaseRTT \tag{1}$$

During congestion avoidance, TCP Vegas compares $Diff$ with two thresholds α and β[3]. TCP Vegas updates its congestion window W_V as follows:

$$W_V = \begin{cases} W_V + \frac{1}{W_V} & if \quad Diff < \alpha \\ W_V - \frac{1}{W_V} & if \quad Diff > \beta \\ W_V & otherwise \end{cases} \tag{2}$$

The slow start mechanism is based on the same concepts of congestion avoidance: TCP Vegas computes $Diff$ and compares it with a unique threshold γ[4]; as long as $Diff$ is less than γ or W_V is less than the slow start threshold, TCP

[1] From now on, we measure the window size in number of packets

[2] Actually, the initial slow start congestion window value depends on the implementation for both TCP Reno and TCP Vegas.

[3] Suggested values are $\alpha = 1$ and $\beta = 3$

[4] Suggested value is $\gamma = 1$.

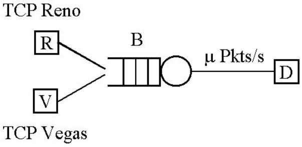


Fig. 1. Network model.

Vegas increases its congestion window by one packet every other round trip delay. Then, TCP Vegas performs congestion avoidance.

After fast retransmit, TCP Vegas reduces the window threshold to the half of the current congestion window and its window by a factor 3/4.

We finally observe that the retransmission mechanisms of TCP Vegas are enhanced with respect to TCP Reno because of the use of a fine-grained clock.

3 The Model

We are interested in analyzing the behavior of the TCP Reno and TCP Vegas mechanisms under the same conditions and with interacting sources exploiting the two protocols.

Therefore, we consider a network scenario with two isolated TCP sources sharing a common route. One source implements the TCP Reno variant, whereas the other the TCP Vegas variant. We assume the sources always have data to send and the transmitted packets have all the same length L.

Along the forward path there is a single bottleneck link with capacity μ packets/sec and a FIFO buffer of size B packets. The others links (including those crossed by the TCP acknowledgments) have a capacity such that they do not limit the source throughput.

The sources experience the same propagation delay T, because they have the same route and destination. For propagation delay T we mean here the time elapsing since the transmission of the packet from the source and the arrival of the corresponding acknowledgment, with the exclusion of the waiting time in the bottleneck buffer. The figure 1 depicts the reference network scenario.

We assume the following hypotheses:

1. The receiver does not pose any limitation to the value of the sender congestion window.
2. The bandwidth-delay product is larger than the buffer size ($\mu T > B$).
3. The sources detect a packet loss via duplicate acknowledgment by using the fast retransmit and fast recovery algorithms. Thus, after observing a packet loss the sources perform the congestion avoidance algorithm.
4. The buffer is empty at the beginning of the congestion avoidance.
5. The sources experience packet loss almost simultaneously; therefore they begin the congestion avoidance together.

These hypotheses lead to a cyclical model for the steady-state behavior of the sources. In figure 2 the periodic evolution of the congestion windows is shown

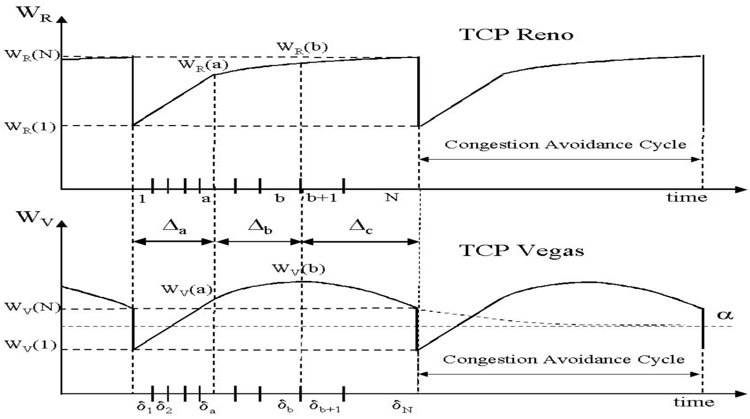


Fig. 2. Congestion Window Evolution of TCP Reno and TCP Vegas.

respectively for TCP Reno (upper plot) and TCP Vegas (lower plot).

Each cycle, called from now on congestion avoidance cycle, begins after a packet loss is detected. During each cycle the congestion avoidance algorithm regulates the window evolution.

By applying the approach already adopted in [9], we consider a congestion avoidance cycle divided in N mini-cycles of duration $\{\delta(i), i = 1, 2, 3, ..., N\}$. The mini-cycle is the time interval during which the same window is maintained. The value of the window during the i-th mini-cycle is denoted by $W(i)$. Formally, the mini-cycles are defined as follows: $\delta(1)$ is T, $\{\delta(i), i = 2, ..., N\}$ is the time elapsing since reception of the acknowledgment of the last packet sent in the window $W(i-1)$ and the arrival of the acknowledgment of the last packet sent in the window $W(i)$.

As for TCP Vegas, we assume in our analysis it has a unique threshold, i.e. $\alpha = \beta$. This simplification does not strongly change the nature of TCP Vegas, because it only eliminates the stable interval without modifying the principles of its mechanisms. Moreover, we can assume that $FlightSize \cong W_V$ and $BaseRTT \cong T$. Therefore, from (2) the congestion window W_V is updated for every incoming acknowledgment as follows:

$$W_V = \begin{cases} W_V + \frac{1}{W_V} & if \quad \left(\frac{W_V}{T} - \frac{W_V}{RTT}\right) \cdot T \leq \alpha \\ W_V - \frac{1}{W_V} & if \quad \left(\frac{W_V}{T} - \frac{W_V}{RTT}\right) \cdot T > \alpha \end{cases} \tag{3}$$

In order to derive the throughput of the sources we must calculate the number of packets sent during a single congestion avoidance cycle and its duration.

First, we calculate the window size of the sources at the end of each cycle, because they affect all the window dynamics.

Let $\{W_V(i), i = 1, ..., N\}$ and $\{W_R(i), i = 1, ..., N\}$ be the size of the i-th congestion window respectively for TCP Vegas and TCP Reno.

To find the final value $W_V(N)$ of the TCP Vegas congestion window, we must consider the behavior of TCP Vegas congestion control.

Let μ_V be the available bandwidth for TCP Vegas:

$$\mu_V = \frac{W_V}{W_R + W_V}\mu \tag{4}$$

If q is the average size of the queue experienced by the packet corresponding to the incoming acknowledgment:

$$W_V + W_R = \mu T + q \tag{5}$$

From (3), TCP Vegas tries to reach an equilibrium in which its congestion window W_V is such that:

$$\left(\frac{W_V}{T} - \frac{W_V}{RTT}\right) \cdot T = \alpha \tag{6}$$

Because W_V/RTT can be interpreted as the available bandwidth μ_V, from (4), (5) and (6), the equilibrium value for the TCP Vegas congestion window is:

$$W_V = \frac{\alpha\,(\mu T + q)}{q} \tag{7}$$

Since at the end of the congestion avoidance phase we can assume $q = B$:

$$W_V(N) = \frac{\alpha\,(\mu T + B)}{B}$$

Since the total number of packets that can be accommodated in the network is $\mu T + B$, we observe a packet loss when $W_V(N) + W_R(N) = \mu T + B + 1$. Hence, the congestion window of the Reno source at the end of the congestion avoidance cycle is:

$$W_R(N) = (\mu T + B)\left(\frac{B - \alpha}{B}\right) + 1$$

Because in the TCP Reno variant the congestion window is halved after detecting a packet loss via triple duplicate acknowledgments, we have for its initial congestion window size $W_R(1)$:

$$W_R(1) = \frac{W_R(N)}{2} \tag{8}$$

Instead, in TCP Vegas the congestion window $W_V(1)$ is generally set to 3/4 of the final congestion window $W_V(N)$. However, since we want to analyze the impact on the TCP Vegas performance of the amount of the window decreasing, we can generically assume:

$$W_V(1) = \lambda \cdot W_V(N) \tag{9}$$

with λ a positive constant less than 1.

Because during congestion avoidance TCP Reno congestion window $W_R(i)$ grows by 1 packet every mini-cycle:

$$W_R(i) = W_R(1) + i - 1 \quad for \quad i = 1, ..., N \tag{10}$$

Thus, the total number P_{Reno} of packets transmitted from the TCP Reno source during a single congestion avoidance cycle is:

$$P_{Reno} = \sum_{i=1}^{N} W_R(i) = N \cdot W_R(1) + \frac{N(N-1)}{2}$$

To calculate the number of packets transmitted by TCP Vegas during a single congestion avoidance cycle and its duration, we can distinguish three phases:

1. From mini-cycle 1 to a: as long as $W_V + W_R \leq \mu T$ we can assume there is no packets in the bottleneck buffer, so $\{\delta(i) = T, i = 1, ..., a\}$. TCP Vegas increases its congestion window by 1 packet every T, because $RTT \cong T$.
2. From mini-cycle $(a+1)$ to b: the duration $\{\delta(i), i = a+1, ..., b\}$ of the mini-cycles increases because of the queuing delay experienced by the transmitted packets; however TCP Vegas still enlarges its window by 1 packet every $\delta(i)$, because it has not reached the equilibrium value in (7) yet.
3. From mini-cycle $(b+1)$ to N: TCP Vegas reduces its congestion window to preserve the equilibrium in (7) while the queuing size increases because of the TCP Reno source.

In the first phase the congestion windows increase for each mini-cycle by one packet and the window growth is identical for the two sources; then, the number a of mini-cycles of duration T is:

$$a = (W_V(a) - W_V(1) + 1) = (W_R(a) - W_R(1) + 1) \tag{11}$$

When the link is saturated, the sum of the windows of the two sources is:

$$W_V(a) + W_R(a) = \mu T \tag{12}$$

From (11) and (12):

$$\begin{aligned} W_V(a) &= \frac{1}{2}\left(\mu T + W_V(1) - W_R(1)\right) \\ W_R(a) &= \frac{1}{2}\left(\mu T + W_R(1) - W_V(1)\right) \\ a &= \frac{1}{2}\left(\mu T - W_R(1) - W_V(1)\right) + 1 \end{aligned}$$

The number of packets sent by TCP Vegas during the first phase and its duration Δ_a are respectively:

$$A_V = \sum_{i=1}^{a} W_V(i) = a \cdot W_V(1) + \frac{a(a-1)}{2} \quad ; \qquad \Delta_a = a \cdot T$$

After the first phase the available bandwidth μ of the bottleneck link is divided between the two sources proportionally to the their respective windows.

Thus, the TCP Vegas available bandwidth $\mu_V(i)$ at the i-th mini-cycle is:

$$\mu_V(i) = \frac{W_V(i)}{W_V(i) + W_R(i)} \cdot \mu \quad for \quad i = a+1, ..., N \tag{13}$$

The duration of each mini-cycle is:

$$\delta(i) = \frac{W_V(i)}{\mu_V(i)} \qquad for \quad i = a+1, ..., N \tag{14}$$

From (13) and (14):

$$\delta(i) = \frac{W_V(i) + W_R(i)}{\mu} \qquad for \quad i = a+1, ..., N \tag{15}$$

where the sum of the windows along the second phase is obtained by considering that the average queuing size increases by two packets every mini-cycle:

$$W_V(i) + W_R(i) = \mu T + 2(i-a) \qquad for \qquad i = a+1, ..., b$$

From (15), the duration Δ_b of the second phase is:

$$\Delta_b = \sum_{i=a+1}^{b} \delta(i) = \left(T + \frac{b-a+1}{\mu}\right)(b-a)$$

The number of packets sent by TCP Vegas during Δ_b is:

$$B_V = (b-a)W_V(a) + \frac{(a+1+b)(b-a)}{2}$$

From (7), TCP Vegas begins to reduce its congestion window when the queuing size q is $2(b-a)$, with b satisfying the following equation:

$$W_V(b) = \frac{\alpha\left(\mu T + 2(b-a)\right)}{2(b-a)} \tag{16}$$

Since $W_V(b) = W_V(a) + b - a$, we calculate b by solving (16):

$$b = a + \frac{-\left(W_V(a) - \alpha\right) + \sqrt{\left(W_V(a) - \alpha\right)^2 + 2\alpha\mu T}}{2}$$

The window size of TCP Reno is $W_R(b) = W_R(a) + b - a$. In the third phase, the queuing size in each mini-cycle can increase at most by 1 packet. Thus, the congestion window of TCP Vegas can follow the equilibrium value given in (7). Thus, by solving (7) with $q = W_V(i) + W_R(b) + i - b - \mu T$, $W_V(i)$ is:

$$W_V(i) = \frac{-C(i) + \sqrt{C(i)^2 + 4\alpha\left[W_R(b) + i - b\right]}}{2} \qquad for \quad i = b+1, ..., N \tag{17}$$

where $C(i) = W_R(b) + i - b - \alpha - \mu T$.

From (15), the duration Δ_c of the third phase is:

$$\Delta_c = \sum_{i=b+1}^{N} \delta(i) = \sum_{i=b+1}^{N} \frac{W_R(i) + W_V(i)}{\mu}$$

with $W_R(i)$ and $W_V(i)$ respectively calculated according to (10) and (17).

The number of packets sent by TCP Vegas during the third phase is:

$$C_V = \sum_{i=b+1}^{N} W_V(i)$$

Therefore, the average throughput Λ_{Vegas} of TCP Vegas is the ratio of the total number P_{Vegas} of transmitted packets to the time Δ:

$$\Lambda_{Vegas} = \frac{P_{Vegas}}{\Delta} = \frac{A_V + B_V + C_V}{\Delta_a + \Delta_b + \Delta_c}$$

The average throughput Λ_{Reno} of TCP Reno is:

$$\Lambda_{Reno} = \frac{P_{Reno}}{\Delta} = \frac{P_{Reno}}{\Delta_a + \Delta_b + \Delta_c}$$

The above analysis is valid if the buffer size is sufficient to allow TCP Vegas to reach the equilibrium status expressed by (7). If this is not the case, we have again the first and the second phases shown if figure 2, whereas the third phase is absent. When the buffer size is less than a threshold B_{th}, TCP Vegas behaves like TCP Reno as already argued in [10], and the average throughput of the two sources only depends on how much TCP Vegas reduces its congestion window after detecting loss, i.e. on the amount of λ. For instance, if $\lambda = 1/2$ the two sources are perfectly equivalent and they experiment the same average throughput. Otherwise, if $\lambda > 1/2$ TCP Vegas pushes more packets than TCP Reno into the pipe. The technique used to calculate the average throughput of the two sources in this case is similar to that adopted above and is shown in Appendix A. The buffer threshold B_{th} is calculated by imposing that the final TCP Vegas congestion window $W_V(N)$ is equal to the congestion window at the equilibrium:

$$W_V(N) = \frac{(\mu T + B_{th} + 1)}{3 - 2\lambda} = \frac{\alpha(\mu T + B_{th})}{B_{th}}$$

Thus, the buffer threshold $B_{th} \cong \alpha(3 - 2\lambda)$.

4 Validation of the Model

In order to validate the analytic model, we carried out simulations under *ns* [11], a network simulator widely used in the networking research community.

The network topology used in the simulations reproduces that shown in figure 1. The two sources start at time 0 and the simulation lasts 600 seconds.

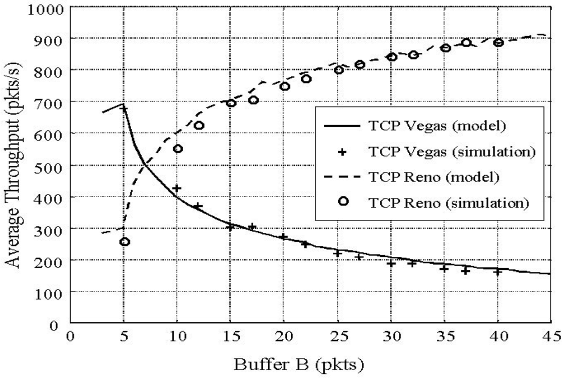


Fig. 3. Model Validation - Varying the buffer size B ($\mu = 1080 pkts/s$, $\alpha = \beta = 3$, $\lambda = 3/4$, $d = 500\ bytes$)

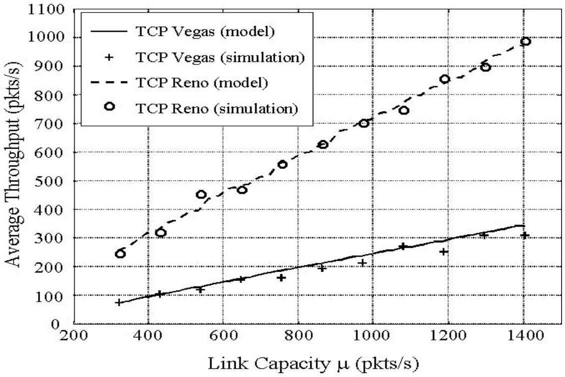


Fig. 4. Model Validation - Varying the link capacity μ (B=20, $\alpha = \beta = 3$, $\lambda = 3/4$, $L = 500\ bytes$)

The source average throughput is calculated over the last 400 seconds, in order to capture the steady-state dynamics. In the simulations the sources exhibit a steady-state periodic behavior, as assumed in our model.

Given the particular *ns* implementation of TCP Vegas, we could not compare our analytic model with the simulations for buffer size between the threshold B_{th} and $[B_{th} + 3(3 - 2\lambda)]$. The reason of this is that the implementation is such that TCP Vegas congestion window oscillates between the equilibrium (7) and the equilibrium plus three packets. This phenomenon is negligible for larger buffers.

In figure 3 we show the model performance obtained by varying the buffer size B. The other parameters in these simulations are fixed to: $\mu = 1080\ pkts/s$, $\alpha = \beta = 3$, $\lambda = 3/4$, $L = 500\ bytes$.

The model captures very well the behavior of the sources. In particular, the dynamics of TCP Vegas are faithfully reproduced with the maximum absolute percentage error of 12.2% (Buffer B=30pkts), whereas as for TCP Reno the maximum absolute percentage error is of 9.9% (Buffer B=5pkts).

As we expected, for small buffers TCP Vegas outperforms TCP Reno. In fact, under the buffer threshold (in this case the threshold B_{th} was 5) TCP Vegas behaves like TCP Reno, because it is not able to reach a stable status. Moreover, TCP Vegas reduces its congestion window after the fast retransmit by $\lambda = 3/4$, whereas TCP Reno halves it.Hence TCP Vegas goes better.

According to the model, the behavior of the sources changes quickly around the buffer threshold B_{th}: it is sufficient to change the buffer from 5 packets to 10 packets to change from a throughput ratio of $\frac{\Lambda_{Vegas}}{\Lambda_{Reno}} = 2.58$ to a throughput ratio of $\frac{\Lambda_{Vegas}}{\Lambda_{Reno}} = 0.76$. Thus, for large buffers TCP Vegas performance quickly deteriorate: its average throughput tends to go to zero.

In figure 4 we show the average throughput of the two sources obtained by varying the bottleneck link capacity μ. The other parameters are fixed in this case to: $B = 20\,pkts$, $\alpha = \beta = 3$, $\lambda = 3/4$, $L = 500\,bytes$.

Also in this case the model captures very well the behavior of the two sources.

According to the model the ratio of TCP Vegas throughput to TCP Reno throughput goes lightly reducing itself when the capacity μ increases. However the slope is too slow to be appreciated in the simulation results.

5 Conclusions

In this paper we presented an analytic model to compute the average throughput of two interactive sources, one implementing as transport protocol the TCP Vegas variant, the other implementing the TCP Reno variant. The aim of the model is to quantitatively establish the vulnerability of TCP Vegas with respect to a concurrent TCP Reno source.

The simulations we carried out show the large accuracy of the model in capturing TCP Vegas and TCP Reno dynamics.

Even if the model reproduces a very simple network scenario with only two sources, it can give interesting indications about TCP Vegas shortcomings.

As further work, our intention is to use the model as a base point to find possible mechanisms to improve TCP Vegas performance in scenarios where there are also TCP Reno sources, in order to stimulate the use of TCP Vegas in the future Internet.

Appendix A

When the buffer B is less than the threshold B_{th}, TCP Vegas and TCP Reno behave in the same way: they increase their congestion window by 1 packet for each mini-cycle until a packet loss occurs. Thus, they have respectively $W_V(N) = W_V(1) + N - 1$ and $W_R(N) = W_R(1) + N - 1$. Since:

$$W_V(1) = \lambda W_V(N); \quad W_R(1) = \frac{1}{2} W_R(N); \quad W_V(N) + W_R(N) = \mu T + B + 1$$

the number N of mini-cycles and the packets transmitted by the sources are:

$$N = \frac{(\mu T + B + 1)(1 - \lambda)}{3 - 2\lambda} + 1$$

$$P_{Reno} = \sum_{i=1}^{N} W_R(i) = \frac{3}{2} N(N-1); \quad P_{Vegas} = \sum_{i=1}^{N} W_V(i) = \frac{1+\lambda}{2(1-\lambda)} N(N-1)$$

We distinguish two phases to calculate the duration of the congestion avoidance cycle: from mini-cycle 1 to mini-cycle a we have $\delta(i) = T$, because we have no packets in the buffer; from mini-cycle $(a+1)$ to N for their duration we must also consider the waiting time in the buffer. Therefore, the duration Δ of the congestion avoidance phase is:

$$\Delta = aT + \frac{1}{\mu}\left[\left(\frac{N-1}{1-\lambda} + N - a + 1\right)(N-a)\right]$$

where a is the number of mini-cycles taken to saturate the bottleneck link :

$$a = \frac{1}{2}\left(\mu T - \frac{N-1}{1-\lambda}\right) + 1$$

References

1. S. Floyd, *A Report on Recent Developments in TCP Congestion Control*, IEEE Communications Magazine, Vol. 9, No. 4, April 2001.
2. L. S. Brakmo, L. L. Peterson, *TPC Vegas: end-to-end congestion avoidance on a global Internet*, IEEE JSAC,Vol.13, No.8, October 1995.
3. J. Mo, R. L. V. Anantharam, J. Walrand, *Analysis and comparison of TCP Reno and Vegas*, Proc. of IEEE Globecom'99, Rio de Janeiro (Brazil), December 1999.
4. Yuan-Cheng Lai, Chang-Li Yao, *The performance comparison between TCP Reno and TCP Vegas*, Proc. of Seventh International Conference on Parallel and Distributed Systems, Iwate (JAPAN), July 2000.
5. U. Hengartner, J. Bolliger and Th. Gross. *TCP Vegas Revisited*, Proc. of IEEE INFOCOM 2000, Tel Aviv (Israel), March 2000.
6. C. Fu et al., *Performance Degradation of TCP Vegas in Asymmetric Networks and its Remedies*, Proc. of ICC2001, Helsinki (Finland), June 11-14, 2001.
7. G. Hasegawa et al., *Analysis and Improvement of Fairness between TCP Reno and Vegas for Deployment of TCP Vegas to the Internet*, Proc. of ICNP, 2000.
8. R. W. Stevens, *TCP/IP Illustrated, Vol I The protocols*, Addison-Wesley, U.S.A., 1994.
9. T.V. Lakshman, U. Madhow, *The Performance of TCP/IP for Networks with High Bandwidth-Delay Product and Random Loss*, IEEE/ACM Transactions on Networking, Vol. 5, No. 3, June 1997.
10. Ait-Hellal, O.; Altman, E., *Analysis of TCP Vegas and TCP Reno*, Proc. of IEEE ICC '97, Vol. 1, Montreal (Canada), June 1997.
11. ns-LBL v.2.1b5, available via http://mash.cs.berkeley.edu/ns/ns.html.

Performance Sensitivity and Fairness of ECN-Aware 'Modified TCP'

Archan Misra[1] and Teunis J. Ott[2]

[1] IBM T J Watson Research Center,
19 Skyline Drive, Hawthorne, NY 10532, USA.
archan@us.ibm.com

[2] Department of Computer Science,
New Jersey Institute of Technology, Newark, NJ 07102, USA.
ott@oak.njit.edu

Abstract. The paper discusses how Explicit Congestion Notification (ECN) can be used to devise a congestion control mechanism for the Internet, which is more rapidly reactive and allows best-effort flows to rapidly adjust to fluctuations in available capacity. Our ECN-mod protocol involves simple modifications to TCP behavior and leverages more aggressive marking-based router feedback. Simulations show that ECN-mod is better than TCP NewReno even for Web-style intermittent traffic sources, and makes the link utilization significantly less sensitive to the variation in the number of active flows. Simulations also show that, while ECN-mod flows obtain a larger portion of the available capacity than conventional best-effort traffic, they do not starve or significantly penalize such TCP-based flows.

1 Introduction

The advantages of using Explicit Congestion Notification (ECN) [1,2] to provide unambiguous congestion feedback to adaptive (TCP) Internet traffic are well documented in literature. We, however, believe that the full benefit of ECN-capable routers has not been effectively realized: *a much more powerful and responsive congestion control framework can be developed if TCP is modified to differentiate between packet marking and packet losses.* When faced with rapid variations in the available bandwidth, such a modified *rapidly reactive* protocol must possess two conflicting characteristics:

- During congestion, the adaptive flows must backoff rapidly to prevent *congestion collapse.*
- Whenever additional bandwidth becomes available, flows should rapidly increase their transmission rate to avoid under-utilization of available capacity.

The design of such an ECN-aware TCP-like protocol for rapid adaptation to variable capacity was presented in [3,4], which suggested that modifications to TCP behavior must be designed in tandem with marking behavior in routers. The protocol modifications exploit the fact that packet *marking* probabilities can

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 624–635, 2002.

be made as high as 100% without causing any undesirable behavior (as opposed to packet *dropping* probabilities which need to be restricted to $\sim 10 - 15\%$.)

In this paper, we first provide a brief recap of our suggested 'ECN-mod' window adaptation algorithm, placing it in the context of a generalized class of 'polynomial' [5] window adaptation algorithms. In particular, we investigate the tradeoffs resulting from a choice of the various coefficients of the polynomial window adjustment procedure. We then report on the result of extensive simulation studies that investigate the properties of our protocol.

Unlike our earlier studies in [4], which used persistent TCP sources, we first use bursty "Web-like" TCP sources and observe the performance of our 'ECN-mod' protocol. As with our earlier observations with persistent TCP sources, we see that in contrast to current ECN-aware TCP NewReno, ECN-mod flows can achieve better link utilization when faced with rapidly changing available bandwidth. However, the improvement in utilization is not as dramatic as with persistent TCP traffic. Using a further set of simulation studies, we show a far more important benefit of the use of the 'ECN-mod' protocol– it makes the network performance much less sensitive to variations in the actual traffic loads. ECN-mod flows (unlike the current ECN-aware TCP flows) are able to operate well even when the marking rates are as high as $\sim 80-90\%$; previous research [6, 7] has clearly documented why such aggressive marking rates may be needed for satisfactory randomized congestion feedback under heavy traffic loads. We also study the 'TCP-friendliness' of our ECN-mod protocol by observing the potential unfairness in resource-sharing between conventional TCP and ECN-mod flows.

2 Generalized Congestion Control and ECN-Mod

Consider an operating environment where an IP flow achieves reliable transmission by using per-packet acknowledgment. Whenever a router recognizes the onset of congestion in its buffer, it sets a "Congestion Experienced" (CE) bit (also called *marking* the packet) in the header of appropriate packets. By having the destination echo this bit in an acknowledgment packet, the source can be informed of such network congestion.

For window-based protocols operating under the TCP paradigm, source adaptation to such congestion can be described by the following generalized behavior: *Whenever an acknowledgment arrives for an unmarked data packet, the congestion window increases from its current value W by $incr(W)$. If, however, the acknowledgment indicates that the data packet had been marked in the forward path, the congestion window is decreased from W by $decr(W)$.*

Note that our framework is much simpler than alternative congestion control models suggested for the Internet (e.g., [8,9]), which are directly concerned with ensuring fairness among competing flows. For example, [8] proposed a rate-based algorithm, where links explicitly update and propagate their shadow congestion costs, and where a source directly adjusts its rate based on its own cost sensitivity. [10] presented the Random Early Marking (REM) algorithm where such shadow costs could be communicated simply by intelligently adjusting the packet marking probability in the network buffers; sources in that scheme, however, adjust their congestion window only periodically. On the other hand, [9] proposed

Charge-Sensitive TCP, where each flow needs to be aware of its instantaneous round-trip delay, transmission rate and congestion window size, and then uses an explicit target window size to regulate the growth of the congestion window. In contrast, our framework does not assume such intelligence at the TCP source, and does not necessarily require the network buffers to dynamically adjust their packet marking function.

For a constant marking probability p, the 'drift' (or the change in the expected value of the congestion window W_{n+1} at the $(n+1)^{th}$ acknowledgment, given the window W_n after the n^{th} acknowledgment) is given by

$$E[W_{n+1} - W_n | W_n = W] = drift(W,p) = (1-p).incr(W) - p.decr(W) =$$

$$p.incr(W).\left(\frac{1-p}{p} - \frac{decr(W)}{incr(W)}\right). \tag{1}$$

Let $q(W)$ be the function

$$q(W) = \frac{decr(W)}{incr(W)}. \tag{2}$$

In that case, if the marking probability p is constant, $q(W)$ will fluctuate around $\frac{1-p}{p}$. The function $q(W)$ is really the *response surface* of the sources to router behavior; as a protocol designer, we can thus first choose $q(.)$ arbitrarily, and then still choose between different values of $incr(.)$ and $decr(.)$ (as long as their ratio remains unchanged). In fact, a legitimate way of designing a congestion protocol is to first choose the response surface $q(W)$ and then directly adapt the window W from an estimate of the marking probability p, without even defining separate $incr(.)$ and $decr(.)$ functions (an approach used in [11]).

For practical reasons, we restrict ourselves to the 'polynomial' class [5] of adaptation algorithms, where

$$incr(w) = c_1 w^{\alpha},\ decr(w) = c_2 w^{\beta}. \tag{3}$$

To ensure that the window does not grow without bound for any given probability, we need $\alpha < \beta$. For the polynomial class of algorithms, this drift would be 0 when $(1-p) * c_1 * W^{\alpha} = p * c_2 * W^{\beta}$. Accordingly, a flow transporting a very large file and subject to a constant marking probability p would observe its congestion window fluctuate around a central value $w(p)$, given by:

$$w(p) = \left(\frac{c_1}{c_2}\frac{1-p}{p}\right)^{\frac{1}{\beta-\alpha}}. \tag{4}$$

2.1 Current TCP Response and Our ECN-Mod Algorithm

Under TCP's current *congestion avoidance* algorithm [13], the congestion window *cwnd* (expressed in terms of the MSS or Maximum Segment Size) increases 1 once every round trip time (RTT) in the absence of congestion; on detection of a congestion episode, *cwnd* decreases from its instantaneous value W by $\frac{W}{2}$. Neglecting transients such as fast recovery and slow-start, TCP's congestion control mechanism is thus a member of the polynomial class, with the parameters

$$TCP: \ c_1 = 1, \ \alpha = -1, \ c_2 = \frac{1}{2}, \ \beta = 1. \tag{5}$$

Of course, most modern TCP versions, such as NewReno or Vegas, halve their window only once for multiple packet losses occurring within a single window (and thus presumably corresponding to a single congestion event).

To provide a more reactive ECN-mod TCP, we use two modifications:

- Make $incr(W)$ more aggressive than TCP, so that it can rapidly increase its *cwnd* in the absence of marking.
- Make $decr(W)$ milder, so that an ECN-mod source can reduce its sending rate in a much more gradual manner.

Of course, to throttle sources rapidly in such an environment, the marking probability for ECN-mod sources should be corresponding higher; more precisely, the buffers should have a higher slope in the marking function. Moreover, our ECN-mod protocol is assumed to respond to *all* ECN-marked packets, even if it leads to multiple reductions within a single window worth of packets. The detailed analysis for the specific choices for β, α, c_1 and c_2 was presented in [3,4], which also recommended an implementation-friendly version of ECN-mod with $\beta = 1, \alpha = 0$.

Since c_1 and c_2 are scaling constants, their choice was more a matter of proper engineering design. We merely require c_2 to be smaller than $\frac{1}{2}$ (current TCP practice) to achieve milder backoff and c_2 to be corresponding small to have reasonable values for the 'average' window size ($w(p)$ in equation (4)) for moderately small p. We have thus experimentally studied a variety the the following members of the ECN-mod family of algorithms

$$ECN-mod: \ c_1 = \{0.625, 0.025\}, \ \alpha = 0, \ c_2 = \frac{1}{8}, \ \beta = 1, \tag{6}$$

which ensure that, for small p, the expected number of marked packets per RTT, lies in the range $(\frac{1}{5}, 5)$.

3 Simulation Parameters and Choices

Our simulation studies are performed using the ns-2 [14] simulator. To simulate a variable-bandwidth environment for best-effort traffic, we used Voice-over-IP (VoIP) sources as higher priority traffic. While each VoIP flow was modeled as per the specifications of the G.711 codec as an exponentially modulated on-off process, the total number of instantaneous calls was modeled as a birth-death process, with call arrival rate λ and exponentially distributed holding times with mean $\frac{1}{\mu}$.

For the graphs plotted here, the best-effort flows were either

- "ECN-aware NewReno", which implements the current TCP algorithm of halving the window in response to both dropped and marked packets.

- "ECN-mod" (or modified ECN), where the source reacts to marked packets as in section II.B and to dropped packets as in TCP NewReno.

We used both a) persistent TCP sources, which involved the transfer of infinite-sized files, and b) Web-TCP sources (using parameters reported in [15]), where a single flow alternates between a *active transfer* phase (during which a new TCP connection is used to transfer a finite-sized file) and an *inactive* phase where the source remains in an idle state.

3.1 Router Marking/Dropping Behavior

Random packet marking and dropping was implemented by a RED [12] queue. The marking function (for ECN NewReno flows), $p(Q)$, was based on the 'gentle' variant [16] of RED and is denoted as $p(Q)$, with the marking probability a linear function of the queue occupancy Q. For ECN-mod flows, the marking function was modified to be more aggressive, such that: *given a queue occupancy Q, the average congestion window size for a best-effort flow was the same for all choices of the congestion window protocol.* Accordingly, the marking function for ECN-mod packets, $p_{mod}(Q)$ is:

$$p_{mod}(Q) = \left(1 + \frac{c_2}{c_1} * \sqrt{\frac{2 * (1 - p(Q))}{p(Q)}}\right)^{-1}, \tag{7}$$

where $p(Q)$ is the basic RED marking function.

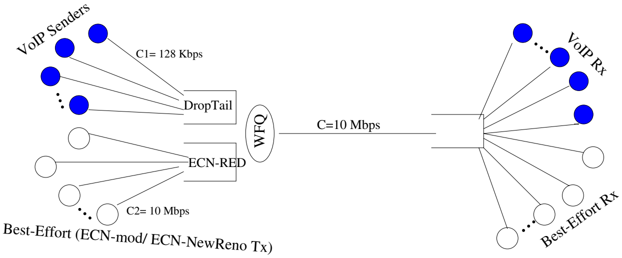


Fig. 1. Simulation Topology for WFQ Experiments

4 Effectiveness and Parameter Insensitivity of ECN-Mod

We now report on simulation studies that investigate how ECN-mod flows perform relative to ECN-NewReno flows, and how their utilization varies with changes in the offered load and marking probabilities. The simulation topology is as shown in Figure 1, with higher-priority (VoIP) and best-effort (TCP) traffic buffered in two separate queues, and Weighted Fair Queuing (more precisely, SCFQ) used to isolate the two classes. To provide voice higher priority, the VoIP class had a weight of 0.8, compared to 0.2 for TCP traffic, even though the offered load of VoIP traffic was often much lower than that of TCP. Admission control is performed for VoIP traffic by having the network block more than $Max.VoIP$ simultaneous voice sessions. For the plots provided here, the bottleneck link capacity C is 10 Mbps; the VoIP queue was sized to have a maximum drain time of 20 msecs. The RED parameters for the best-effort queue (in packets) had $min_{th} = 25$, $max_{th} = 75$ and buffer size $B = 150$ (following the recommendations in [16]). The RTT of the best-effort connections are uniformly spaced out over the interval $(25, \ldots, 250)$ msecs.

4.1 Performance Improvement with Web TCP Traffic

The effect of ECN-mod in increasing the link utilization and TCP throughput for persistent TCP traffic was presented in [4]. In this subsection, we thus focus on the network performance when the sources are not persistent but rather represent finite-sized Web-based file transfers (using the Barford-Crovella model). Figures 2 and 3 plot the simulation results (averaged over 10 runs) when N, the number of Web TCP sources, equals 150 and $p_{max} = 0.2$. Figure 2 plots the total goodput (VoIP+ TCP), as well as the TCP goodput alone as the average number of simultaneous VoIP calls is varied (by varying λ). It is easy to see that ECN-mod and ECN-NewReno do not exhibit significant differences, although ECN-mod (for well-chosen values of c_1) does achieve slightly better utilization than ECN-NewReno. The reason for this is easy to understand: while the dynamic variation in the number of active TCP transactions will cause transient network congestion, the long-term TCP throughput does not change since the best-effort traffic is essentially source-constrained. More importantly, unlike earlier studies with persistent TCP traffic, setting $c_1 = 0.125$ in ECN-mod performs better than $c_1 = 0.0625$. Since most Web file transfers usually complete during the initial slow-start transient (before congestion feedback is even activated), a more aggressive choice of the window increase coefficient typically leads to higher TCP goodput. Of course, as in [4], an over-aggressive value of c_1 can increase the queue variability significantly, leading to buffer underflow and loss of network utilization[1].

Figure 3 plots the packet marking rates for the best-effort flows and the coefficient of variation (defined as $\frac{Std.Deviation}{Mean}$) of the best-effort queue occupancy. As expected, the marking rates turn out to be higher for ECN-mod than ECN-NewReno– our congestion control framework is based on a more aggressive congestion notification ability in Internet routers. (The packet loss rate on all these runs was essentially 0, indicating that congestion control was achieved solely via packet marking). More importantly, the coefficient of variation for ECN-mod flows is lower (sometimes by as much as 30%) than that of ECN-NewReno flows. Since ECN-mod flows exhibit a much milder backoff than ECN NewReno, the occupancy of the RED queue (for good choices of ECN-mod parameters) fluctuates in a much smoother fashion, leading to much smaller coefficients of variation than that with ECN-NewReno. Accordingly, while the use

[1] This result suggests an interesting possibility of having c_1 during the initial slow-start transient different from the subsequent value of c_1 during the congestion avoidance phase; we do not, however, explore this idea further in this paper.

of the ECN-mod window adjustment protocol may not improve the long-term network utilization significantly (since the Web traffic load is essentially source-constrained), it does lead to better the network dynamics, such as a smoother queue evolution and lower packet jitter.

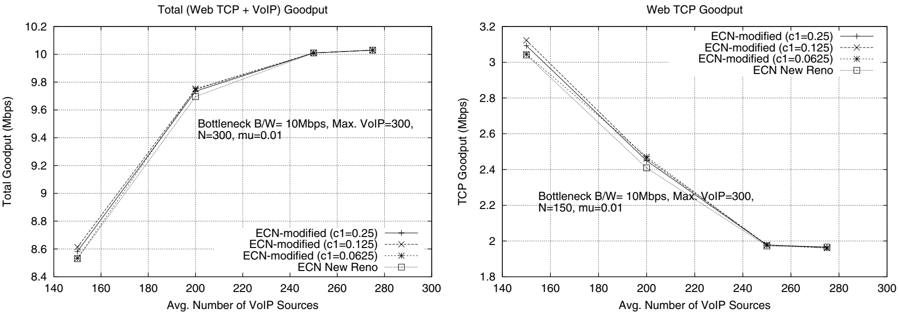


Fig. 2. Comparative Capacity Utilization for Web Sources

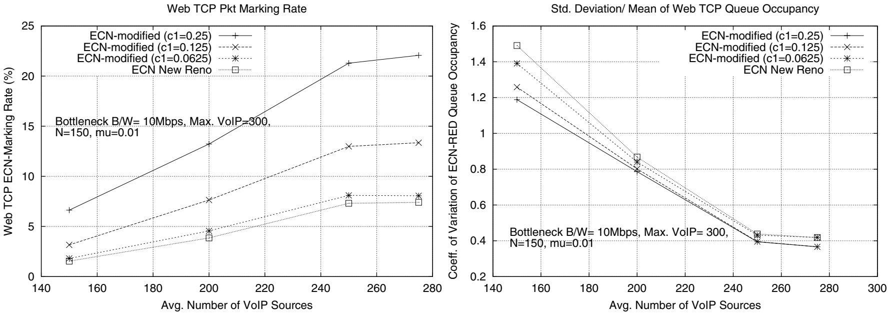


Fig. 3. TCP Marking Rates and Queue Variability for Web Sources

4.2 Sensitivity to Load Variation

We now consider the performance of ECN-mod vs. ECN-NewReno as the number of *persistent* best-effort flows is varied. (Results with Web TCP sources are qualitatively similar and omitted due to space constraints.) For these studies, the average number of VoIP flows was kept constant at 200, by setting $\lambda = 2.0$. We varied the number of best-effort (persistent) flows, N, from $10 - 200$. The graphs study two interesting settings of p_{max}, namely 0.2 and 1.0.

We first consider the commonly used RED setting of $p_{max} = 0.2$. Figure 4 shows the variation in the total (VoIP+best-effort), as well as the best-effort, goodput as N is varied from 10 to 200. We see that, when N is relatively large, both ECN-mod and ECN-NewReno obtain comparable goodput. However, when N is small, ECN-mod ($c_1 = 0.0625$) clearly outperforms ECN-NewReno, since ECN-mod is able to utilize the available bandwidth more aggressively. Figure

5 shows the variation in the packet dropping and marking rates respectively. We see that, as N increases, the packet loss rates become very high (as large as $\sim$ 10% for $N = 200$). This indicates that a p_{max} setting of 0.2 does not provide sufficiently strong feedback to prevent undesirable packet losses under high loads– a larger value of p_{max} is preferred.

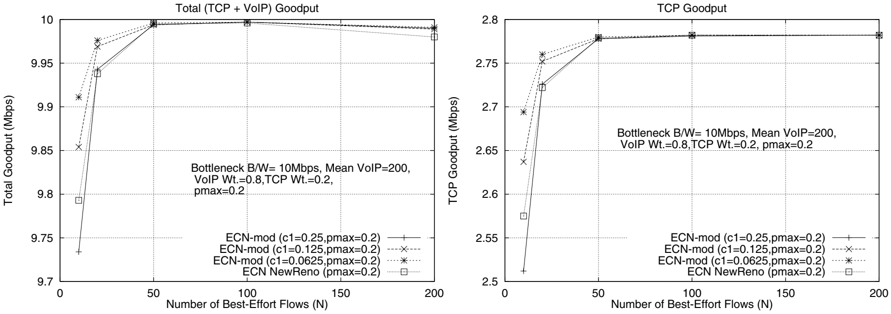


Fig. 4. Comparative Goodput ($p_{max} = 0.2$)

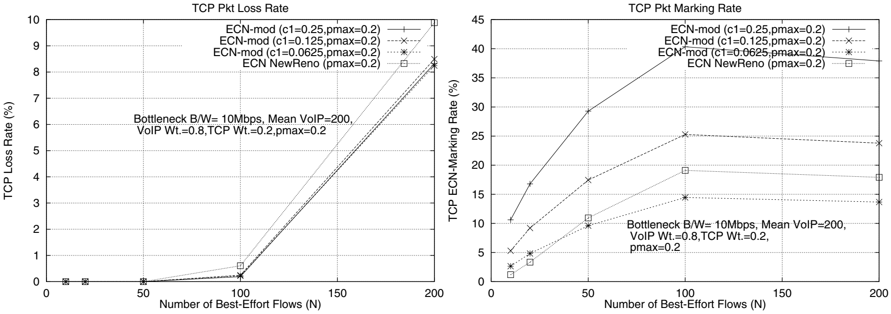


Fig. 5. Comparative Drop/Marking Rates ($p_{max} = 0.2$)

In Figures 6 and 7, we investigate precisely such an aggressive setting, where $p_{max} = 1$. Figure 6 shows the total and best-effort traffic goodput as N is varied. As before, we see that ECN-mod is better than ECN-NewReno in utilizing the available bandwidth. More importantly, as N is increased beyond 50, we see that, while the ECN-mod flows ($c_1 = 0.0625$) always achieve high goodput, the ECN-NewReno goodput actually decreases. This occurs because TCP's policy of halving the window size does not work well at the relatively high marking rates (see Figure 7) obtained when $p_{max} = 1.0$ and N is large. Thus, *the current response of TCP to ECN marking does not allow best-effort flows to operate in environments where routers exhibit aggressive marking behavior.*

The first graph in Figure 7 plots the average marking rates for best-effort traffic, when $p_{max} = 1.0$. Observe that the marking rates for ECN-mod traffic are as high as $\sim$ 85%; yet, Figure 6 shows no degradation in ECN-mod performance. The second graph plots the coefficient of variation of the queue occupancy as N

is increased. While ECN-mod always results in a lower queue variability (smaller coefficient of variation) than ECN-NewReno, the difference is more pronounced for large N, where ECN-NewReno cannot cope with the high marking rates. It is well-known that RED's inability to adaptively vary p_{max} leads to performance degradation as the number of TCP flows is varied. Accordingly, ECN-mod appears to provide the significant advantage of making the best-effort utilization largely independent of the number of active flows; by setting p_{max} to a high value and using the ECN-mod algorithm, we can make the network utilization uniformly high for a very wide range of N and avoid undesirable packet drops.

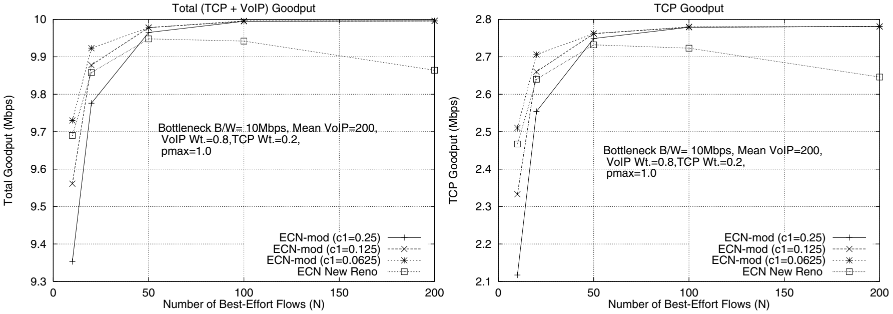


Fig. 6. Comparative Goodput ($p_{max=1.0}$)

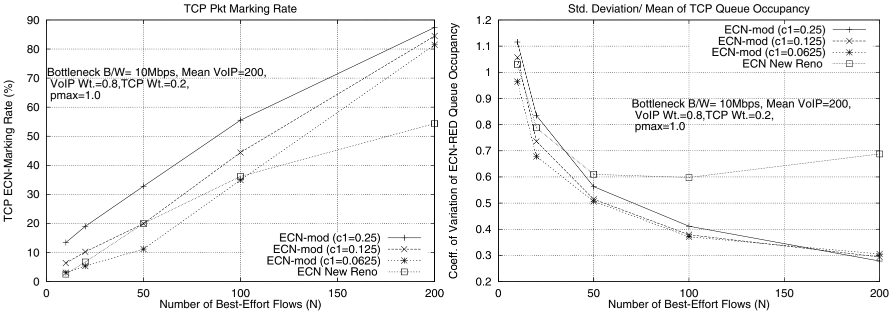


Fig. 7. Comparative Marking/Queue Variation ($p_{max=1.0}$)

5 Fairness between ECN-Mod and TCP NewReno

Since it is impractical to expect that all sources to change their window-adjustment behavior overnight, we have also studied the "TCP-friendliness" of ECN-mod traffic, i.e., the relative sharing of the best-effort bandwidth between competing ECN-mod and conventional TCP flows. To this end, we performed simulations where the router port uses a single FIFO buffer, and the best-effort flows were either all NewReno or ECN-mod or an equal mix of both. The network topology is thus similar to that of Figure 1, except that VoIP no longer has explicit protection through the Class Based WFQ mechanism. For these experiments, the link capacity $C = 10$ Mbps, $min_{th} = 20$, $max_{th} = 60$, $p_{max} = 0.2$ and buffer size $B = 120$. As before, due to space limitations, we report only on experiments with persistent TCP sources.

Figure 8 shows the variation in the total goodput, as well as the TCP goodput, for the various TCP adaptation algorithms when the total number of best-effort sources, N, equals 20. We can clearly see that ECN-mod, with $c_1 = 0.0625$, outperforms the current ECN-NewReno procedure. Once again, we observe that ECN-mod with $c_1 = 0.25$ performs worse than the current ECN-NewReno algorithm, indicating that an overly aggressive choice of parameters may incur severe performance penalties. We also observed that the VoIP throughput was essentially unaffected by the best-effort traffic, since the non-adaptive UDP flows do not react to the 'marking' of packets at the buffer.

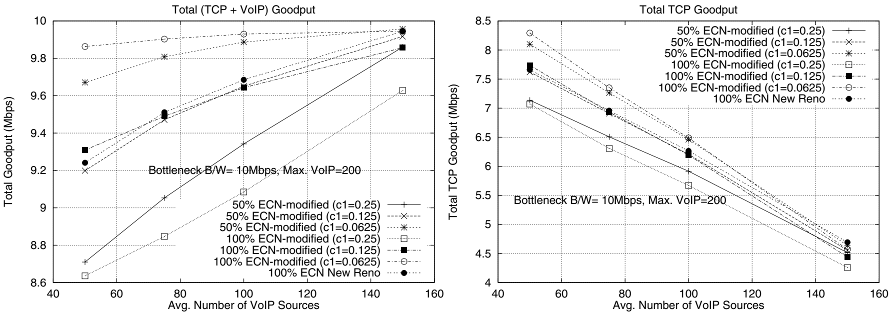


Fig. 8. Throughput for Mixed NewReno/ECN-mod Traffic

Figure 9 first plots the relative throughputs of the ECN-mod and ECN-NewReno flows, when the best-effort traffic consists of an equal number (10 each) of ECN-mod and ECN-NewReno sources. Clearly, while ECN-mod has the higher goodput (for $c_1 = 0.0625$), ECN-NewReno sources are not completely shut out and obtain about $20\% - 25\%$ less goodput than their ECN-mod counterparts. A far more important point can be observed by studying the second graph, which studies the goodput achieved by the 10 ECN-NewReno sources, when the other 10 sources were either ECN-NewReno or ECN-mod. It is interesting to see that, for certain loads, the 10 ECN–NewReno sources obtain higher goodput if the other sources are ECN–mod ($c_1 = 0.0625$) than if they are ECN–NewReno. This illustrates the important point that, under certain circumstances, *the performance of conventional ECN-NewReno sources is improved (in absolute terms) in the presence of other ECN-mod traffic sources, even though, relatively speaking, the ECN-NewReno sources receive a smaller fraction of the total goodput.* This clearly mitigates any potential fairness concern, since the overall increase in the utilization levels swamps the reduction in ECN-NewReno's share of the total achieved goodput. Additional plots (omitted here due to space constraints) further show that ECN-mod (with $c_1 = 0.0625$) flows result in a lower coefficient of variation of the queue occupancy than corresponding ECN-NewReno flows.

In this case, where VoIP packets are buffered in the same queue, this directly translates into *smaller delay jitter* for individual VoIP packets.

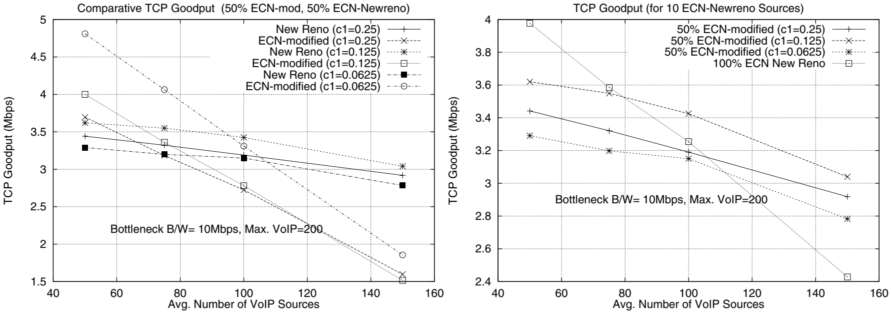


Fig. 9. Relative Goodput for ECN-NewReno and ECN-mod Sources

We have also studied the fairness issues between ECN-mod and non-ECN capable TCP flows using the same setup. As expected, ECN-unaware flows perform worse than ECN-capable ECN-mod flows. However, those studies also demonstrate that ECN-mod, while capturing a relatively larger portion of the available bandwidth, *never leads to starvation or significant deterioration* of the conventional TCP flows. Such studies indicate that it may be possible for ECN-mod and conventional TCP flows to co-exist in the network, especially if the network buffers are able to apply a more aggressive marking behavior selectively to the ECN-mod flows.

6 Conclusions

In this paper, we continue our investigation of a rapidly reactive congestion control framework for adaptive (best-effort) TCP-like flows. This framework includes an ECN-mod protocol that has a more aggressive decrease and milder decrease than conventional TCP, and requires routers to mark packets much more aggressively than currently envisioned. Simulation studies indicate the performance benefits of ECN-mod over ECN-NewReno, demonstrated earlier for persistent TCP sources, apply even when the flows transfer finite-sized files and are source-constrained. In particular, the use of ECN-mod window adaptation leads to smoother buffer behavior and less drastic variation in the instantaneous total traffic loads. We, however, need to be conservative in the choice of ECN-mod coefficients: if the window increase coefficient is too large, network utilization may drop significantly.

Further studies also show that the use of the ECN-mod protocol makes the link utilization by adaptive traffic significantly less sensitive to the number of active flows, and the precise setting of RED's p_{max} parameter. Studies using a mixture of ECN-mod flows and conventional TCP flows also demonstrate that ECN-mod does not significantly penalize conventional TCP traffic; while ECN-mod does grab a larger share of the available bandwidth, it also improves the overall utilization. While not intended to be conclusive, our results do argue that the current TCP behavior, of responding to the notification of an ECN-marked packet in exactly the same way as it reacts to the discovery of a lost packet ([1,2]), may be sub-optimal. The best shape of the marking function, however, remains an open question.

References

1. Floyd, S.: TCP and Explicit Congestion Notification, *ACM Computer Communications Review* 21 no 5, 1994.
2. Ramakrishnan, K.K., Floyd, S.: A Proposal to add Explicit Congestion Notification(ECN) to IP, *RFC 2481*, January 1999.
3. Ott, T.J.: ECN Protocols and the TCP Paradigm, *ftp://ftp.research. telcordia.com/pub/tjo/ECN.ps*, 1999.
4. Misra, A., Ott, T.J.: Jointly Coordinating ECN and TCP for Rapid Adaptation to Varying Bandwidth, *Proceedings of IEEE MILCOM 2001*, to appear, October 2001.
5. Bansal, D., Balakrishnan, H.: Binomial Congestion Control Algorithms, *Proceedings of IEEE INFOCOM*, April 2001.
6. Ott, T., Lakshman, S., Wong, L.: SRED: Stabilized RED, *Proceedings of IEEE INFOCOM*, March 1999.
7. Feng, W., Kandlur, D., Saha, D. and Shin, K.: Adaptive Packet Marking for Providing Differentiated Services in the Internet, *Proceedings of ICNP'98*, October 1998.
8. Kelly, F.P., Maulloo A.K., Tan D.K.H.: Rate Control for Communication Networks: Shadow Prices, Proportional Fairness and Stability, *Journal of the Operational Research Society*, March 1998.
9. La, R.J., Anantharam V.: Charge-Sensitive TCP and Rate Control in the Internet, *Proceedings of IEEE INFOCOM*, March 2000.
10. Low, S.H., Lapsley D.E.: Optimization Flow Control, *IEEE/ACM Transactions on Networking*, December 1999.
11. Floyd, S., Handley M., Padhye J., Widmer J: Equation-based Congestion Control for Unicast Applications, *Proceedings of ACM SIGCOMM 2000*, Sept 2000.
12. Floyd, S., Jacobson V., Random Early Detection Gateways for Congestion Avoidance, *IEEE/ACM Transactions on Networking*, August 1993.
13. Jacobson, V., Karels. K.: Congestion Avoidance and Control, *Proceedings of ACM SIGCOMM*, 1988.
14. The ns-2 network simulator, *http://www-mash.CS.Berkeley.EDU/ ns.*
15. Barford M., Crovella M.: Generating Representative Workloads for Network and Server Performance Evaluation, *Boston University Technical Report*, BU-CS-97-006, 1997.
16. Floyd, S.: Recommendations on using the 'gentle' variant of RED,
http://www.aciri.org /floyd/red.html, March 2000.

Call Admission Control for 3G CDMA Networks with Differentiated QoS

Qian Huang[1], Hui Min Chen[1], King Tim Ko[2], Sammy Chan[2], and
King Sun Chan[3]

[1] School of Communication and Information Engineering, Shanghai University
No. 149, Yan Chang Road, 200072 Shanghai, China
hqian@online.sh.cn
[2] Department of Electronic Engineering, City University of Hong Kong
Tat Chee Avenue, Kowloon, Hong Kong
{eektko, eeschan}@cityu.edu.hk
[3] Department of Electrical and Electronic Engineering, The University of Hong Kong
Pokfulam Road, Hong Kong
kschan@eee.hku.hk

Abstract. A call admission control (CAC) scheme for 3G wireless networks based on the estimated interference and differentiated quality of service (QoS) is proposed in this paper. In this CAC scheme, the interferences introduced by the to-be-admitted new call and its impact to the QoS of the existing connections in the adjacent cells are considered in call admission. The purpose is to obtain QoS balance amongst neighboring cells within a cluster. Comparison of our proposed scheme with the current scheme without considering the effects of the to-be-admitted calls indicates that the proposed scheme performs better in outage and blocking probabilities.

1 Introduction

The third-generation (3G) network will support broadband applications like streaming video, web browsing and network games via wireless IP [1]. As the radio link is bandwidth limited and error prone, wireless IP applications face some major challenges in practice. QoS is a crucial issue in delivering the services. To guarantee the QoS for different services over the radio link, admission control schemes that can handle differentiated QoS and efficiently utilize the scarce bandwidth are important.

Some admission control schemes for the CDMA based networks have been studied in [2]–[5]. The CAC schemes presented in [2] and [3] were similar: a new call accessed the network with its target signal-to-interference ratio (SIR) requirement; both of the proposed schemes accepted the new call in terms of the estimated residual capacity over the radio link. However, since these two schemes handled the case of single type of voice traffic, the homogeneous QoS requirement on SIR was only considered in CAC.

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 636–647, 2002.

More recent studies on CAC schemes suitable for a system with different services are made in [4] and [5]. These CAC schemes are based on the total tolerable interference in a cell. The CAC scheme presented in [4] focused on the single cell system model, in which the interference from the adjacent cells was neglected. Shen and Ji [5] considered a cell cluster system in their proposed CAC scheme. When a new call arrived, the interference levels from other cells and the path loss experienced by the new call were estimated; if the total equivalent bandwidth taken up by all users in the cell, including the to-be-admitted new call, plus the estimated inter-cell interference exceeded the maximal channel capacity, the new call will be blocked, otherwise, it will be admitted.

However, the scheme in [5] did not consider the QoS degradation and unbalance in the adjacent cells caused by the admitted calls in the serving cell. Particularly, if a new call with high bit rate is admitted at the edge of its serving cell and, its adjacent cells are heavily loaded at the same time, the on-going connections in these adjacent cells may experience QoS degradation and communication outage because of the increase in the interference level. Generally, the on-going connections outage is more unacceptable than the blocking of a new call. In order to reduce the outage probability of the on-going connection and obtain balanced QoS within the neighboring cells, we propose here a new CAC scheme in which the interference introduced by the to-be-admitted call is considered during call admission in both the serving and adjacent cells. A new call will be admitted, if the total estimated resultant interferences in both serving and adjacent cells are under a predefined threshold, so that the guaranteed QoS of the existing connections within the cell cluster will not be violated after the new call is admitted.

In Section 2, the model for our proposed call admission scheme is presented. Section 3 derives the outage and blocking probabilities of the proposed admission control scheme for different service types. Section 4 gives the numerical results and comparisons in performances with an existing admission scheme. The final section gives the concluding remarks.

2 System Model

In this paper, we consider the scenario of differentiated QoS provisioned over a cluster of cells system, in which the cell of interest is surrounded by H adjacent cells ($H = 6$). Services are classified in terms of the requirements of BER and transmission rate. Two typical IP traffics are assumed in our proposal. One is the 8 kbps packet voice, the other is the WWW traffic with average rate equal to 144 kbps. For the packet voice traffic, the ON-OFF model is used. For the WWW traffic, the traffic model proposed in [9] is adopted. The size of a web page has a log-normal distribution, and the thinking interval has an exponential distribution. The call arrival is assumed to follow Poisson distribution for both voice and data services.

The multi-code mode is adopted in our model to support different bit rate services. After a call is admitted, more than one code is allocated for the aggregated transmission rate. Each code is equivalent to a sub-channel, the high speed data service is transported simultaneously over several sub-channels and the equal spreading gain is achieved on each sub-channel. Let m_d be the number of sub-channels for a call, let R_b be the basic bandwidth of the sub-channel, thus the aggregate transmission rate equals to $R_b \cdot m_d$.

It is assumed that the background noise is additive white Gaussian noise (AWGN). For the total interference levels from the voice and data users in the neighboring cells, they are also approximated by the Gaussian distribution as in [6]. With only a few high rate users, the interference distribution might not be Gaussian, however we are making the above assumption for the common distribution of high and low rate users within cells. Let I_v^i, I_d^i, I_v^h and I_d^h respectively be the interference introduced to the cell i and h from the voice and data users in their adjacent cells, their mean and variance can be obtained according to the path loss model used in [6]:

$$\text{Path loss} = 10^{(\xi/10)} r^{-4}, \tag{1}$$

where r is the distance from mobile station to the target BS, ξ is approximated as a Gaussian random variable with zero mean and 8dB variance. For a mobile station (MS) j, assuming its serving BS is i, and adjacent BS is h. Let the received power of MS j at BS i be represented as S_v for the voice service and S_d for the data service. From Equation (1), we can obtain that the interference power level received at BS h from MS j is $S_v \frac{l_j(h)}{l_j(i)}$ and $S_d \frac{l_j(h)}{l_j(i)}$ for the voice and data call respectively, where $l_j(h)$ represents the path loss from the BS h to MS j, $l_j(h) = 10^{(\xi_h/10)} r_{jh}^{-4}$; and $l_j(i)$ represents the path loss from MS j to its serving BS i, $l_j(i) = 10^{(\xi_i/10)} r_{ji}^{-4}$.

We also assume that perfect power control is implemented on the up-link, and equal power can be received at the BS from each user of the same traffic class. Let SIR_{th}^v be the target SIR for the voice service, and SIR_{th}^d for the data service, the target SIR values are corresponding to the BER requirements. Let R_v be the peak bit rate for the voice, and R_d be the average bit rate for the data, $R_d = m_d R_b$. The product of the target SIR and the transmission rate is defined as the equivalent bandwidth requested by the differentiated QoS, that is $B_v = SIR_{th}^v R_v$ for the voice service, and $B_d = SIR_{th}^d R_d$.

Given the radio link bandwidth (w), the maximum total acceptable interference density (I_0) within a CDMA cell, including the inter-cell interference, proves 10 times as high as that of the background noise (N_0), that is $I_0 = 10N_0$ [7]. Let a_j^v represent the voice user j's activity factor, its mean is denoted as α_v, α_v is assumed to be 3/8 [6]; let a_j^d be the activity factor of user j with WWW traffic, and α_d be the mean. Based on the above predefined threshold, when a new call with transmission rate R_{new} and target SIR SIR_{th}^{new} arrives at its serving BS i, the CAC scheme will check whether the interferences measured at BS i and the adjacent BS h ($h = 1, 2, \ldots, H$) violate the following two admission conditions: For BS i, the condition is

$$B_{new} + \sum_{j=1}^{N_v^i} a_j^v B_v + \sum_{j=1}^{N_d^i} a_j^d B_d + \frac{I_v}{I_0} + \frac{I_d}{I_0} \leq 0.9w, \tag{2}$$

where $B_{new} = SIR_{th}^{new} R_{new}$ is the equivalent bandwidth requested by the new call; N_v^i and N_d^i respectively represent the number of existing voice and data calls in the serving cell i; a_j^v and a_j^d represent the activity factor of the j^{th} voice call and data call in cell i; I_v and I_d denote the interference powers from the voice and data calls in the adjacent cells; the levels of I_v and I_d are approximated by a Gaussian distribution.

For BS h, the interference power introduced by the signal power of the new call received at the BS i (S_{new}^i) to the BS h can be obtained by the ratio of the path-loss from mobile station (MS) j to BS h ($l_j(h)$) to that from MS j to BS i ($l_j(i)$), then the admission condition is

$$\sum_{j=1}^{N_v^h} {a'}_j^v B_v + \sum_{j=1}^{N_d^h} {a'}_j^d B_d + \frac{S_{new}^i}{I_0}\frac{l_j(h)}{l_j(i)} + \frac{I_v}{I_0} + \frac{I_d}{I_0} \leq 0.9w, \tag{3}$$

where N_v^h and N_d^h respectively indicate the number of existing voice and data calls in the adjacent cell h; ${a'}_j^v$ and ${a'}_j^d$ represent the activity factor of the j^{th} voice call and data call in cell h. When both Equation (2) and (3) are satisfied, the new call is accepted; otherwise, it is blocked.

3 Derivation of Outage and Blocking Probabilities

In the CDMA-based network, the probability of the received SIR lower than the target SIR is defined as the outage probability, which indicates the QoS performance supported by the network, while the blocking probability is related to the network capacity. Here, we use these two parameters as the criteria in assessing our admission control scheme.

The CDMA reverse link model for calculating the SIR by received voice users is given by Gilhousen *et al.*[6]. Based on this single traffic model, Ayyagari and Ephremides developed a model for integrated voice and data services in [8]. In our analysis, the reverse link model proposed in [8] is adopted. The difference is that the interference caused by the to-be-admitted call to its adjacent cells is included in our model.

The call arrivals for both voice and data services follow Poisson processes with mean arrival rate λ_v and λ_d respectively; let x and y denote the number of arrived calls during the average service time T_s for the voice and data respectively. Let SIR_i^v be the received SIR at the serving BS i from a voice call, and SIR_i^d for the data call, the received SIR for an on-going voice and data call in the serving cell i can be derived as follows:

$$SIR_i^v = \frac{w/R_b}{\sum_{j=1}^{N_v^i - 1} a_j^v + \sum_{j=1}^{N_d^i} a_j^d m_d \frac{S_d}{S_v} + x + y m_d \frac{S_d}{S_v} + \frac{I_v^i + I_d^i + N_0 w}{S_v}} \tag{4}$$

$$SIR_i^d = \frac{w/R_b}{\sum_{j=1}^{N_v^i} a_j^v \frac{S_v}{S_d} + \sum_{j=1}^{N_d^i - 1} a_j^d m_d + (m_d - 1) + x\frac{S_v}{S_d} + y m_d + \frac{I_v^i + I_d^i + N_0 w}{S_d}} \tag{5}$$

Considering the interference from the to-be-admitted call in serving cell i, the received SIR at BS h, can be expressed as follows:

$$SIR_h^v = \frac{w/R_b}{\sum_{j=1}^{N_v^h - 1} a'^v_j + \sum_{j=1}^{N_d^h} \frac{a'^d_j m_d S_d}{S_v} + \sum_{j=1}^{x} \frac{l_j(h)}{l_j(i)} + \sum_{j=1}^{y} \frac{l_j(h)}{l_j(i)} \frac{m_d S_d}{S_v} + \frac{I_v^h + I_d^h + N_0 w}{S_v}} \tag{6}$$

$$SIR_h^d = \frac{w/R_b}{\sum_{j=1}^{N_v^h} a'^v_j \frac{S_v}{S_d} + \sum_{j=1}^{N_d^h - 1} a'^d_j m_d + (m_d - 1) + \sum_{j=1}^{x} \frac{l_j(h) S_v}{l_j(i) S_d} + \sum_{j=1}^{y} \frac{m_d l_j(h)}{l_j(i)} + \frac{I_v^h + I_d^h + N_0 w}{S_d}} \tag{7}$$

In the above equations, the mean and the variance of I_v^i/S_v and I_d^i/S_d can be derived as in [6]:

$$E(I_v^i/S_v) \leq 0.247 N_v^i \cong \mu_v \tag{8}$$

$$E(I_d^i/S_d) \leq 0.247 N_d^i m_d \frac{\alpha_d}{\alpha_v} \cong \mu_d \tag{9}$$

$$var(I_v^i/S_v) \leq 0.078 N_v^i \cong \sigma_v^2 \tag{10}$$

$$var(I_d^i/S_d) \leq 0.078 N_d^i m_d^2 \frac{\alpha_d}{\alpha_v} \cong \sigma_d^2 \tag{11}$$

where α_v and α_d are the means of voice and data activity factor respectively.

From Equation (6) and (7), the interference levels received at the BS h from the to-be-admitted x and y calls in the cell i, are equivalent to the increase of the user number of x and y in the cell h. Let $I'_v/S = \sum_{j=1}^{x} \frac{l_j(h)}{l_j(i)} + \frac{I_v^h}{S}$, $I'_d/S = \sum_{j=1}^{y} \frac{l_j(h)}{l_j(i)} m_d + \frac{I_d^h}{S}$, then we have,

$$E(I'_v/S_v) \leq 0.247(N_v^h + \lambda_v T_s) \cong \mu'_v \tag{12}$$

$$E(I'_d/S_d) \leq 0.247(N_d^h + \lambda_d T_s) m_d \frac{\alpha_d}{\alpha_v} \cong \mu'_d \tag{13}$$

$$var(I'_v/S_v) \leq 0.078(N_v^h + \lambda_v T_s) \cong \sigma_v'^2 \tag{14}$$

$$var(I'_d/S_d) \leq 0.078(N_d^h + \lambda_d T_s) m_d^2 \frac{\alpha_d}{\alpha_v} \cong \sigma_d'^2 \tag{15}$$

From Equation (4) and (5), we can derive the outage probability for the differentiated QoS at the serving BS i:

$$\begin{aligned} P_{i\,out}^{\,v} &= Pr(SIR_i^v \leq SIR_{th}^v) \\ &= Pr\left(\sum_{j=1}^{N_v^i-1} a_j^v + \sum_{j=1}^{N_d^i} a_j^d m_d \frac{S_d}{S_v} + x + y m_d \frac{S_d}{S_v} + \frac{I_v^i + I_d^i + N_0}{S_v} \geq \frac{w/R_b}{SIR_{th}^v}\right) \\ &= Pr\left(x + y m_d \frac{S_d}{S_v} + \frac{I_v^i + I_d^i}{S_v} \geq \frac{w/R_b}{SIR_{th}^v} - \frac{N_0}{S_v} - k_v - k_d m_d \frac{S_d}{S_v}\right) \\ &\quad \cdot \left[\sum_{k_d=0}^{N_d^i} \sum_{k_v=0}^{N_v^i-1} \binom{N_d^i}{k_d} \binom{N_v^i-1}{k_v} \alpha_v^{k_v} (1-\alpha_v)^{N_v^i-1-k_v} \alpha_d^{k_d} (1-\alpha_d)^{N_d^i-k_d}\right] \\ &= \sum_{k_d=0}^{N_d^i} \sum_{k_v=0}^{N_v^i-1} \binom{N_d^i}{k_d} \binom{N_v^i-1}{k_v} \alpha_v^{k_v} (1-\alpha_v)^{N_v^i-1-k_v} \alpha_d^{k_d} (1-\alpha_d)^{N_d^i-k_d} \\ &\quad \cdot Q\left(\frac{\mu_1}{\sqrt{\sigma_1^2 + \sigma_2^2}}\right) \end{aligned} \tag{16}$$

where,

$$\mu_1 = \frac{w/R_b}{SIR_{th}^v} - \frac{N_0}{S_v} - k_v - k_d m_d - (\mu_v + \mu_d \frac{S_d}{S_v}) - \lambda_v T_s - \lambda_d T_s m_d \frac{S_d}{S_v} \tag{17}$$

$$\sigma_1^2 = \lambda_v T_s + \sigma_v^2 \tag{18}$$

$$\sigma_2^2 = (\lambda_d T_s m_d^2 + \sigma_d^2)(\frac{S_d}{S_v})^2 \tag{19}$$

$$\begin{aligned} P_{i\,out}^{\,d} &= Pr(SIR_i^d \leq SIR_{th}^d) \\ &= Pr(\sum_{j=1}^{N_v^i} a_j^v \frac{S_v}{S_d} + \sum_{j=1}^{N_d^i-1} a_j^d m_d + (m_d - 1) + x \frac{S_v}{S_d} \\ &\quad + y m_d + \frac{I_v^i + I_d^i + N_0}{S_d} \geq \frac{w/R_b}{SIR_{th}^d}) \\ &= \sum_{k_d=0}^{N_d^i-1} \sum_{k_v=0}^{N_v^i} \binom{N_d^i-1}{k_d} \binom{N_v^i}{k_v} \alpha_v^{k_v} (1-\alpha_v)^{N_v^i-k_v} \alpha_d^{k_d} (1-\alpha_d)^{N_d^i-1-k_d} \\ &\quad \cdot Q\left(\frac{\mu_2}{\sqrt{\sigma_3^2 + \sigma_4^2}}\right) \end{aligned} \tag{20}$$

where,

$$\mu_2 = \frac{w/R_b}{SIR_{th}^d} - \frac{N_0}{S_d} - k_v \frac{S_v}{S_d} - k_d m_d - (m_d - 1) - (\mu_v \frac{S_v}{S_d} + \mu_d) - \lambda_v T_s \frac{S_v}{S_d} - \lambda_d T_s m_d \tag{21}$$

$$\sigma_3^2 = (\lambda_v T_s + \sigma_v^2)(\frac{S_v}{S_d})^2 \tag{22}$$

$$\sigma_4^2 = \lambda_d T_s m_d^2 + \sigma_d^2 \tag{23}$$

Similarly, from Equation (6) and (7), we can obtain the outage probability at the adjacent BS h:

$$\begin{aligned}
P_{h\,out}^{\;v} &= Pr(SIR_h^v \leq SIR_{th}^v) \\
&= Pr\left(\sum_{j=1}^{N_v^h-1} a'^{v}_{j} + \sum_{j=1}^{N_d^h} a'^{d}_{j} m_d \frac{S_d}{S_v} + \frac{I'_v + I'_d + N_0}{S_v} \geq \frac{w/R_b}{SIR_{th}^v}\right) \\
&= \sum_{k_d=0}^{N_d^h} \sum_{k_v=0}^{N_v^h-1} \binom{N_d^h}{k_d} \binom{N_v^h-1}{k_v} \alpha_v^{k_v}(1-\alpha_v)^{N_v^h-1-k_v} \alpha_d^{k_d}(1-\alpha_d)^{N_d^h-k_d} \\
&\quad \cdot Q\left(\frac{\mu_3}{\sqrt{\sigma_v'^2 + \sigma_d'^2(\frac{S_d}{S_v})^2}}\right)
\end{aligned} \tag{24}$$

where,

$$\mu_3 = \frac{w/R_b}{SIR_{th}^v} - \frac{N_0}{S_v} - k_v - k_d m_d \frac{S_d}{S_v} - (\mu'_v + \mu'_d \frac{S_d}{S_v}) \tag{25}$$

$$\begin{aligned}
P_{h\,out}^{\;d} &= \sum_{k_d=0}^{N_d^h-1} \sum_{k_v=0}^{N_v^h} \binom{N_d^h-1}{k_d} \binom{N_v^h}{k_v} \alpha_v^{k_v}(1-\alpha_v)^{N_v^h-k_v} \alpha_d^{k_d}(1-\alpha_d)^{N_d^h-1-k_d} \\
&\quad \cdot Q\left(\frac{\mu_4}{\sqrt{\sigma_v'^2(\frac{S_v}{S_d})^2 + \sigma_d'^2}}\right)
\end{aligned} \tag{26}$$

where,

$$\mu_4 = \frac{w/R_b}{SIR_{th}^d} - \frac{N_0}{S_d} - k_v \frac{S_v}{S_d} - k_d m_d - (m_d - 1) - (\mu'_v \frac{S_v}{S_d} + \mu'_d) \tag{27}$$

In the above equations, $Q(\cdot)$is the error function with $Q(z)=\frac{1}{\sqrt{2\pi}}\int_z^{\infty} e^{-t^2/2}dt$.

The blocking probability can be derived from the admission conditions given by Equation (2) and (3). The blocking probability for a new call equals to the probability that the admission conditions are violated. We define the admission probability at the serving BS i as P_s^i, and P_s^h for the adjacent BS h. From Equation (2) and (3), we have,

$$P_s^i = Pr(B_{new} + \sum_{j=1}^{N_v^i} a_j^v B_v + \sum_{j=1}^{N_d^i} a_j^d B_d + \frac{I_v^i}{I_0} + \frac{I_d^i}{I_0} \leq 0.9w) \tag{28}$$

$$P_s^h = Pr(\sum_{j=1}^{N_v^i} a'^v_j B_v + \sum_{j=1}^{N_d^i} a'^d_j B_d + \frac{S_{new}^i}{I_0}\frac{l_j(h)}{l_j(i)} + \frac{I_v^h}{I_0} + \frac{I_d^h}{I_0} \leq 0.9w) \tag{29}$$

Then, the blocking probability for a new call in its serving cell i can be expressed as follows:

$$P_{blocking}^i = 1 - (P_s^i \prod_{h=1}^{H} P_s^h) \tag{30}$$

If the new call is the voice service, B_{new} in the above two admission probability expressions should be $B_v = SIR_{th}^v R_v$; for a data service call, it is replaced by $B_d = SIR_{th}^d R_d$.

4 Numerical Results

We compare the performance of the CAC method (denoted as $Scheme1$)which is similar to the proposal in [5] without consideration on the interference brought by the to-be-admitted call to the adjacent cells, with our admission control scheme (denoted as $Scheme2$). In the following comparison, 8 kbps voice service and 144 kbps (average rate) WWW traffic are considered. The average service time (T_s) for both types of calls is assumed to be 100 seconds. A minimal target SIR of 7dB is required by the voice calls to guarantee 10^{-3} BER. We denote R_b as the basic bandwidth of the sub-channel, $R_b = 16kbps$ is used in our analysis. For the voice traffic with 8 kbps bit rate, one sub-channel is sufficient to support two voice channels. For simplification, Gaussian approximation method is used in both schemes to estimate the interference from the adjacent cells.

Initially, we assume that both voice and data calls in the network have identical BER requirement of 10^{-3}, which corresponds to the target SIR, and all the neighboring cells have identical traffic loading. In addition, a predefined outage probability of 5% is given for both voice and data calls. Fig. 1 and Fig. 2 respectively present the simulation results of the capacities in both serving and adjacent cells under different admission control $Scheme1$ and $Scheme2$. Comparing to the results of $Scheme1$ results shown in Fig. 1, Fig. 2 shows that the system using $Scheme2$ goes into stable status in both serving and adjacent cells after 200 seconds. It also shows that given a predefined outage probability, more voice and data calls are supported by $Scheme1$ than $Scheme2$. This is because those to-be-admitted calls which cause more interference to its neighboring cells, are most likely to be rejected in our proposed scheme. Thus, the interference brought by these "bad" calls to the existing connections in both serving and adjacent cells is reduced as much as possible, and the available radio bandwidth is efficiently allocated to support those calls which have lower transmission power under power control and good radio link quality, consequently, the total capacity of the cell cluster using $Scheme2$ is larger than that of $Scheme1$.

Fig. 3 presents the comparison of voice call outage probabilities in both schemes versus the total call arrival rates $\lambda_v + \lambda_d$, when data calls access the network with different BER requirements of 10^{-2}, 10^{-3} and 10^{-6}. The results

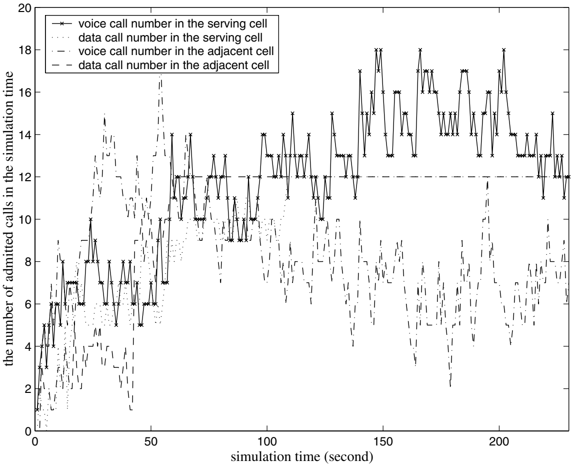


Fig. 1. The system capacity of Scheme 1

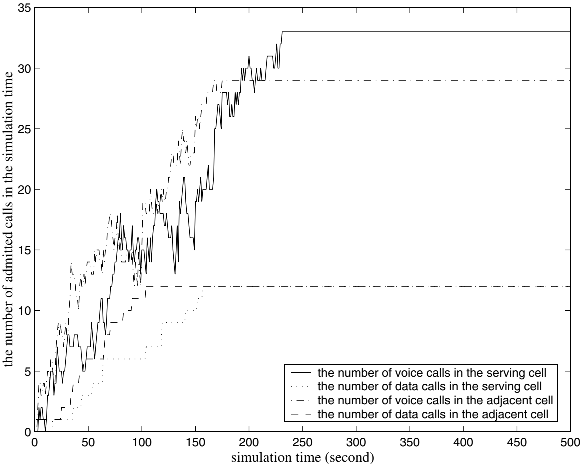


Fig. 2. The system capacity of Scheme 2.

show that the voice call outage probabilities in $Scheme2$ under different data BER requirements are all lower than those of $Scheme1$. The reason is that the to-be-admitted calls generated in the adjacent cells are prevented from introducing unacceptable interference to the serving cell, thus the outage probability in our scheme is reduced. Similar performance is also obtained for the data calls. Fig. 3 also shows that high-rate data calls have a significant impact on the QoS of the low-rate voice calls. As the data BER requirement varies from 10^{-2} to 10^{-6}, the voice call outage probability increases. It shows that admitting data calls with higher QoS requirements will degrade the QoS of the other services, because more interferences will be introduced to other services from each sub-channel of the data call.

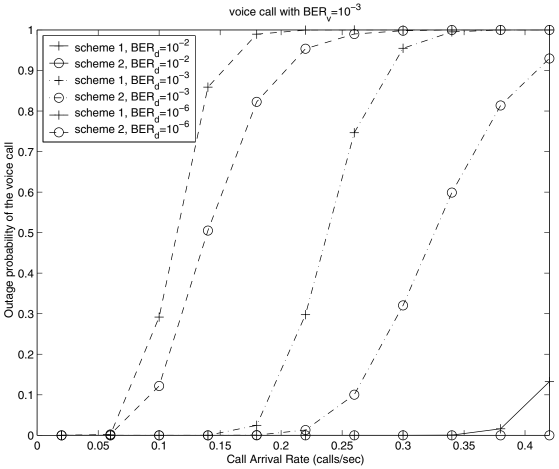


Fig. 3. Comparison of the voice call outage probability in the serving cell.

Fig. 4 presents the call blocking probability in *Scheme*1 and *Scheme*2, when both voice and data calls access the network with BER equal to 10^{-3}. The results in Fig. 4 show that *Scheme*2 outperforms the call blocking probabilities in both types of services. In *Scheme*1, similar call blocking probabilities are expected by both voice and data calls, but the outage probabilities for both types of calls have been increased as in Fig. 3. *Scheme*2 gives a higher call blocking probability to the data calls when the network is heavily loaded, while the outage probability of the voice call is lower than that of *Scheme*1 in Fig. 3. Since the to-be-admitted calls which could introduce unacceptable interference to the neighboring cells are rejected, the total interference measured within the radio link bandwidth is minimized. Accordingly, given the same blocking probability, *Scheme*2 can support larger number of high-rate data calls with the guaranteed QoS than *Scheme*1.

Finally, we consider the case of different traffic loading amongst the cells. We assume that the traffic loading in the serving cell is twice as high as that in the adjacent cells. Under this assumption, the outage probabilities for different offered loads are compared between *Scheme*1 and *Scheme*2 in Fig. 5. Fig. 5 shows that the outage probabilities in *Scheme*1 are higher, but the values of *Scheme*2 are approaching zero in both hot spot (serving cell) and adjacent cells. This is because *Scheme*2 can prevent the increase in interference from the admitted “bad” calls in the adjacent cells, so as to balance the outage probabilities amongst the cells in a cluster and avoid serious QoS deterioration in the heavily loaded cells.

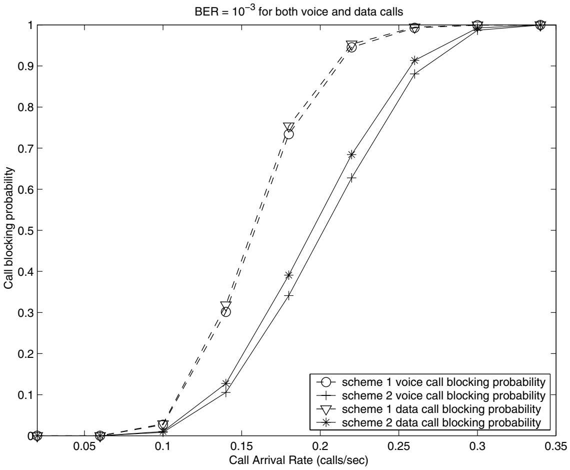


Fig. 4. Comparison of the call blocking probability.

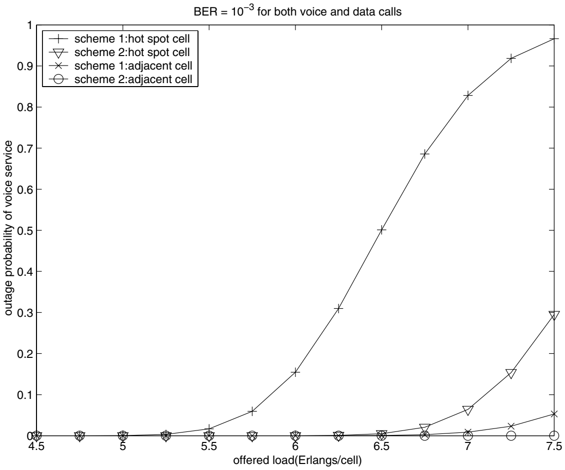


Fig. 5. Outage probability of the voice call under different traffic loading.

5 Conclusion

The admission control scheme proposed in this paper for 3G mobile networks is based on the estimated interferences and differentiated QoS requirements for different services. In order to achieve the optimization of QoS over the radio access network, the inter-cell interference and the impacts caused by the to-be-admitted calls to the QoS of the existing connections in the adjacent cells are also considered in our scheme. Under our proposed admission control scheme, the higher admission probability is given to those calls close to their serving BS or experiencing lower path-loss. On the other hand, the calls locating at the edge of the serving cell and having large path-loss to the serving BS will introduce exceeded interference to adjacent cells. This kind of calls have a higher blocking probability. Comparisons of the proposed scheme with another scheme which does not take the effects of the to-be-admitted calls into account indicates that the proposed scheme performs better in outage and blocking probabilities.

In our scheme, the issue of call hand-off or terminal mobility is not considered. Further study is under way to take this issue into consideration for admission control in a CDMA based 3G system.

Acknowledgment. The work described in this paper was supported by the City University of Hong Kong under Strategic Research Grant 7001124.

References

1. Patel G., Dennett S.: The 3GPP and 3GPP2 Movements Toward an All-IP Mobile Network. IEEE Pers. Commun., Vol. 7, Issue 4. **8** (2000) 62–64
2. Liu Z., Zarki M. E.: SIR Based Call Admission Control for DS–CDMA Cellular Systems. IEEE J. Select. Areas Commun., Vol. 12, No. 4. **5** (1994) 638–644
3. Kim I. M., Shin B. C., Lee D. J.: SIR-Based Call Admission Control by Intercell Interference Prediction for DS-CDMA Systems. IEEE Communications Letters, Vol. 4, No. 1. **1** 2000 29–31
4. Comaniciu C., Mandayam N. B.: QoS Guarantees for Third Generation (3G) CDMA System via Admission and Flow Control. Proceedings, IEEE Veh. Technol. Conf.. (2000) 249–256
5. Shen D., Ji C.: Admission Control of Multimedia Traffic for Third Generation CDMA Network. Proceedings, IEEE INFOCOM 2000. **3** (2000) 1077–1086
6. Gilhousen K. S., Jacobs I. M., Padovani R., Viterbi A. J., Weaver L. A., Wheatley C. E.: On the Capacity of a Cellular CDMA System. IEEE Trans. Veh. Technol., Vol. 40, No. 2. **5** (1991) 303–312
7. Viterbi A. M., Viterbi A. J.: Erlang Capacity of a Power Controlled CDMA System. IEEE J. Select. Areas Commun., Vol. 11, No. 6. **8** (1993) 892–990
8. Ayyagari D., Ephremides A.: Cellular Multicode CDMA Capacity for Integrated (Voice and Data) Services. IEEE J. Select. Areas Commun., Vol. 17, No. 5. **5** (1999) 928–938
9. Molina M., Castelli P., Foddis G.: Web Traffic Modeling Exploiting TCP Connections' Temporal Clustering through HTML-REDUCE. IEEE Network. **5/6** (2000) 46–55

Performance Evaluation of Channel Switching Scheme for Packet Data Transmission in Radio Network Controller

Yoshiaki Ohta, Kenji Kawahara, Takeshi Ikenaga, and Yuji Oie

Dept. of Computer Science and Electronics, Kyushu Institute of Technology
Iizuka, Fukuoka 820-8502, Japan
yoshiaki@infonet.cse.kyutech.ac.jp, {kawahara, ike, oie}@cse.kyutech.ac.jp

Abstract. W-CDMA (Wideband-CDMA) is expected for the radio access technology of the third-generation mobile telecommunication systems. In the second-generation systems, voice traffic from each user has mainly been transmitted via the dedicated transport (radio) channel. In addition, the third-generation systems will efficiently accommodate data traffic based on the packet transmission in the *shared* common transport channel. Therefore, data traffic can be transmitted via one of two types of channels: i.e., dedicated channel and common channel. However, the channel selecting/switching scheme in RNC (Radio Network Controller) has not been standardized. Thus, in the present paper, we will propose some channel switching schemes and evaluate their performance in terms of the packet loss probability and the utilization of dedicated channels by means of simulations.

Keywords: IMT-2000, W-CDMA, RNC, Dedicated Channel, Common Channel

1 Introduction

Wireless communications have been recently attracted and spread widely with the rapid growth of the Internet. In the second-generation mobile telecommunication systems, the major services are limited to basic services such as voice, facsimile, and low-rate-data transmission. In the third-generation mobile telecommunication systems, a variety of services such as high speed Internet access, multimedia data transmission, and global roaming will be expected. For that reason, the ITU (International Telecommunication Union) began its studies on a global standard for mobile telecommunication systems, which is referred to as IMT-2000 (International Mobile Telecommunications-2000) [1][2][3]. In the third-generation mobile telecommunication systems, the future radio transmission technology is strongly expected to efficiently transmit not only legacy voice traffic but also the data traffic based on the packet transmission. Thus, a lot of proposals for radio transmission technology candidates have been submitted to the ITU. Most of them were based on CDMA (Code Division Multiple Access) but with differences in technical details, and to prevent multiple standard problems, they have been integrated and developed to some global standards. Especially in them, W-CDMA (Wideband-CDMA) [4][5] receives much attention for the radio transmission technology and most research has been done in this area. The standard of W-CDMA is defined in detail by 3GPP (3rd Generation Partnership Project) [6], and throughout the standard process, it discussed the packet transmission (radio) channel structure and proposed one method. In

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 648–659, 2002.

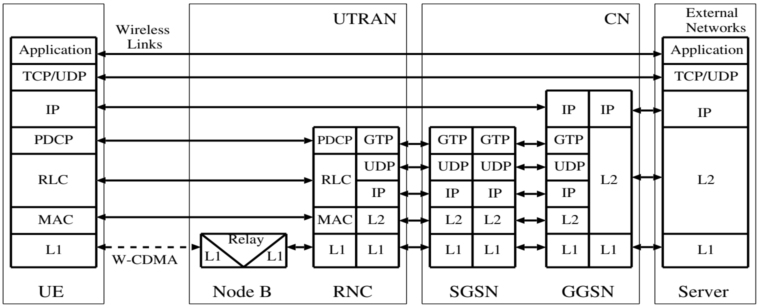


Fig. 1. Protocol stacks of W-CDMA packet data services

the method, two channels are mainly provided for the packet transmission: dedicated channel and common channel. For example, in the case that a large amount of traffic is transmitted by some flow, a channel will be dedicated to it and its packets can be efficiently transmitted over it, whereas flows with only a small amount of traffic share the common channel. Radio resources can be efficiently utilized in some adaptive way of selecting an appropriate transmission channel according to the traffic characteristics [7]. The channel selecting/switching is controlled by RNC (Radio Network Controller), and the specific scheme in RNC has not been standardized.

Therefore, our major interest in the present paper is to clarify the issues related to channel selecting/switching schemes and study their characteristics. We will thus propose specific channel selecting/switching schemes in accordance with current W-CDMA specifications and evaluate their performance in terms of the packet loss probability and the utilization of dedicated channels by means of simulations. In addition, we will discuss their characteristics based upon performance comparison.

2 Architecture of RNC

Fig. 1 shows an example of protocol stacks in W-CDMA packet data services [6]. Several nodes exist between Internet servers and UE (User Equipment) or mobile terminals. In CN (Core Network), two distinct elements exist: GGSN (Gateway General packet radio service Support Node) and SGSN (Serving General packet radio service Support Node). GGSN is the switch at the point where nodes are connected to external networks. All incoming and outgoing packets must go through GGSN. SGSN is the database that serves UE in its current location, and provides the function of packet switching and routing. In UTRAN (Universal Terrestrial Radio Access Network) which handles radio-related functions, two distinct elements also exist: RNC and Node B. RNC is responsible for managing radio resources in wireless links. Node B corresponds to a base station (BS) which handles the radio communication over W-CDMA air interfaces. In these protocol stacks, RNC includes the physical layer and the link layer, and it segments packets received from external networks into several data blocks of fixed length, recovers transmission errors that occurred in wireless links, and assigns proper transmission channels. These functions are provided in RLC (Radio Link Control)/MAC (Media Access Control) layers described in RNC.

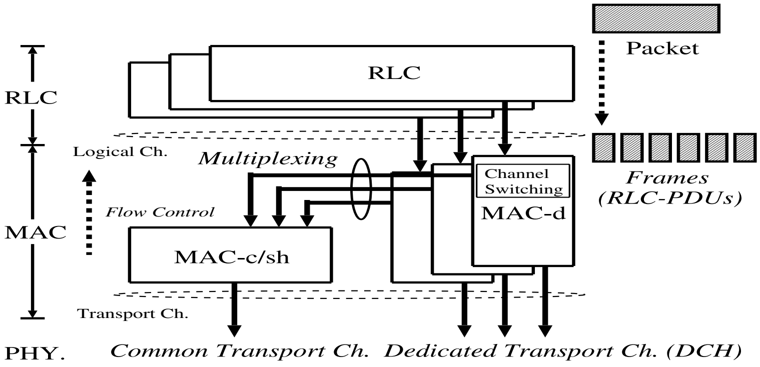


Fig. 2. The architecture of RLC/MAC layers

2.1 Architecture of RLC/MAC Layers

Fig. 2 illustrates the architecture of RLC/MAC layers as well as a data transmission method used in it. In the RLC layer, packets transmitted from external networks are segmented into RLC PDUs, i.e., link layer frames, they are then forwarded to the RLC dedicated buffer for each flow. Thus, the transmission unit in RLC/MAC layers is the frame, to which control information such as its sequence number is added in the header. To achieve good performance even in high error rate links, an ARQ (Automatic Repeat reQuest) mechanism based on the frame is provided for protecting against transmission errors through a limited number of retransmission attempts. An ARQ protocol is implemented in W-CDMA packet data services. This protocol is based on the selective repeat scheme.

In the MAC layer, mainly two sublayers are described: MAC-d sublayer and MAC-c/sh sublayer. The main function of the MAC-d sublayer is channel selecting/switching and control of dedicated channels. The main function of the MAC-c/sh sublayer is control of common channels. One MAC-d sublayer in RNC is allocated for each UE, while only one MAC-c/sh in RNC is shared by all UE in a cell.

In RLC/MAC layers, two types of data channels are described: logical channel and transport one. The logical channel resides between RLC and MAC layers, while the transport one is located between MAC and physical layers. Furthermore, two types of data channels are also described in the transport channel: dedicated channel and common one. After frames are transmitted from the RLC layer, they first arrive at each MAC-d sublayer via a logical channel, and they are then transmitted to each corresponding UE via one of two types of transport channels. In the case that the dedicated channel is used, they are successively forwarded to each corresponding UE. On the other hand, when the common one is used, RNC schedules their transmission to multiplex them into one *shared* channel, and then forwards them in turn to each corresponding UE.

Thus, when frames are transmitted via the common channel, the congestion of the MAC layer would more frequently happen since they are multiplexed into one *shared* channel. When RNC recognizes it by means of feedback information sent from the MAC layer, it controls the transmission rate of the RLC layer to avoid the congestion. The functions of RLC/MAC layers are described in [8][9].

2.2 Proposed Channel Switching Scheme

In the channel selecting/switching method described in [9], a transmission channel is selected by RNC adaptively based upon the queue length of the RLC dedicated buffer; two thresholds, upper threshold (THU) and lower threshold (THL), are employed for that purpose. The detail is described in the following:

- If the queue length of the RLC dedicated buffer exceeds the predetermined upper threshold THU, the transmission channel is switched from the common channel to the dedicated one.
- If the queue length of the RLC dedicated buffer falls below the predetermined lower threshold THL, the transmission channel is switched from the dedicated channel to the common one.

According to the above method, we propose specific channel selecting/switching schemes in the following. The following schemes first assign the common channel for frame transmission to gain the statistical multiplexing effect. However, they are different from each other in how to begin transmitting frames on the dedicated channel after selecting it instead of the common one.

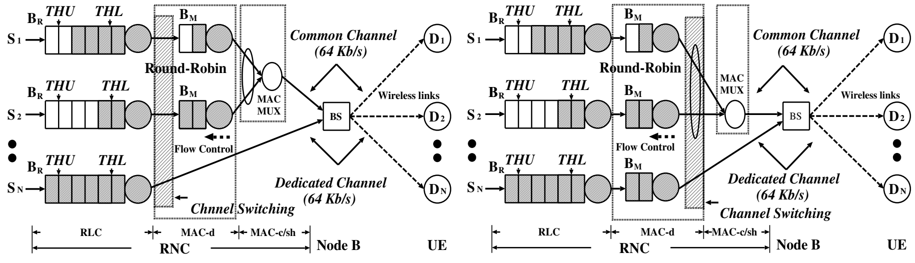
- *Scheme 1*: Frames currently stored in the RLC dedicated buffer are kept waiting until all frames stored in the MAC-d dedicated buffer are transmitted on the common channel. This assumes that the MAC-d dedicated buffer for each flow is equipped to multiplex frames on the common channel so that those frames cannot be transmitted on the dedicated one.
- *Scheme 2*: Frames currently stored in the RLC dedicated buffer are immediately transmitted regardless of whether there are any frames stored in the MAC-d dedicated buffer or not. This scheme is also based upon the above assumption. There are possibilities that the related flow uses both the common channel and dedicated one at the same time until all frames in the MAC-d dedicated buffer are transmitted.
- *Scheme 3*: Frames currently stored in the MAC-d dedicated buffer are immediately transmitted on the dedicated channel assigned now, and then frames stored in the RLC dedicated buffer are successively forwarded to the MAC-d dedicated one. This scheme is free from the above assumption. Hence, the channel switching entity follows the MAC-d dedicated buffer.

3 Simulation Model

In this section, we describe our model for simulation to evaluate the channel selecting/switching schemes presented in Sec. 2. We add some modifications to Network Simulator NS Version 2 developed by VINT Project [10] and use it for our research.

3.1 RLC/UE Model

Fig. 3 shows the simulation model of RLC layers and UE; Fig. 3(a) illustrates the model for Scheme 1 and 2, while Fig. 3(b) describes that for Scheme 3. In RNC, each RLC layer is equipped with a dedicated buffer of B_R [frame] and each MAC-d sublayer is equipped with a dedicated buffer of B_M [frame]. The buffer size B_R and B_M is fixed



(a) Scheme 1, Scheme 2

(b) Scheme 3

Fig. 3. Simulation model for channel selecting/switching schemes

at 10 and 2, respectively. One common channel of 64 Kb/s is used here, in which the round-robin scheduling algorithm is employed to multiplex frames in several MAC-d dedicated buffers. In this case, to avoid the congestion of the MAC layer, the following flow control scheme based on the queue length of the MAC-d dedicated buffer is adopted:

```
if (MAC-d queue length >= 1)
        RLC transmission rate = 0 (Kb/s)
else if (MAC-d queue length < 1)
        RLC transmission rate = 64 (Kb/s)
```

Note that by adopting this scheme, the MAC-d buffer overflow never occurs. Furthermore, several dedicated channels are employed, each of which is of 64 Kb/s.

3.2 Traffic Model

We assume that each traffic source S_1–S_N transmits web traffic to D_1–D_N, where N is the number of traffic sources. Each source generates traffic according to on/off process. The duration for which frames are successively transmitted is denoted by T_{ON} [frame]. It follows an exponential distribution and the mean length of which is 10 frames of 42 bytes in size. We assume that retransmitted frames are included in T_{ON}. The duration for which frames are not transmitted is denoted by T_{off} [frame] and it varies according to the amount of traffic denoted by λ; i.e., $\lambda = N \times T_{ON}/(T_{ON} + T_{OFF})$. Note that the total amount of traffic $\lambda = 1.0$ indicates a traffic of 64 Kb/s.

The traffic model is limited so that there may be difficulties in deriving general conclusions. Additionally, we do not consider the wireless aspect of the system in terms of the channel error. However, we here focus on the channel selecting/switching schemes, and our major purpose is to get their fundamental performance.

4 Numerical Results and Discussions

In this section, we will evaluate the performance of proposed three schemes for channel selecting/switching by using our simulation model presented in the previous section. For each of these schemes, we first investigate the impact of two thresholds THU and THL

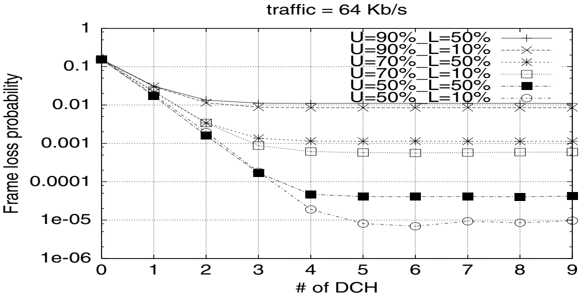


Fig. 4. Frame loss probability (Scheme 1)

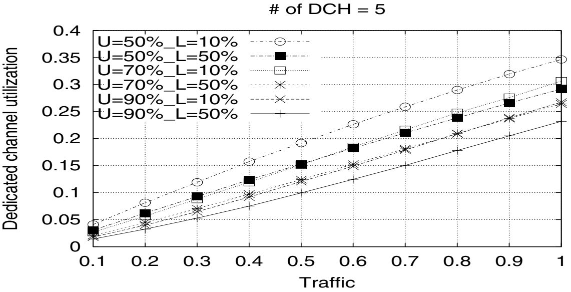


Fig. 5. Dedicated channel utilization (Scheme 1)

of the RLC dedicated buffer on the frame loss probability as a function of the maximum number of dedicated channels, which can be utilized by all traffic sources. We then investigate the utilization of the dedicated channel, which is defined by the ratio of the number of frames transmitted via dedicated channels to that of all frames received at each UE. The aim of the proposed schemes is to effectively transmit frames via the common channel and to decrease the utilization of the dedicated channel. Furthermore, by showing the impact of the number of sources and the amount of traffic on the performance, we discuss the optimal number of dedicated channels.

4.1 Evaluation of Scheme 1

In this subsection, we focus on the performance of the channel switching Scheme 1. We set the total amount of traffic arriving at Node B, λ, to 1.0, namely, 64 Kb/s and the number of traffic sources, N, to 10.

Fig. 4 shows the frame loss probability of the RLC dedicated buffer. From this figure, we can find that the frame loss probability increases with threshold THU. In this scheme, even if the queue length of the RLC dedicated buffer for a flow exceeds THU and the dedicated channel is then assigned to the flow, they will be never transmitted until all frames stored in the corresponding MAC-d dedicated buffer are transmitted via the common channel; this is HOL (Head-of-Line) blocking effect. Therefore, if THU is set to a large value, most newly arriving frames at the RLC dedicated buffer will be lost.

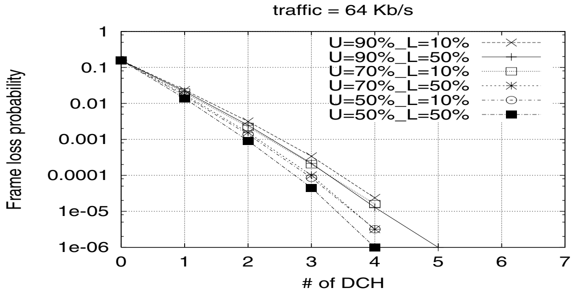


Fig. 6. Frame loss probability (Scheme 2)

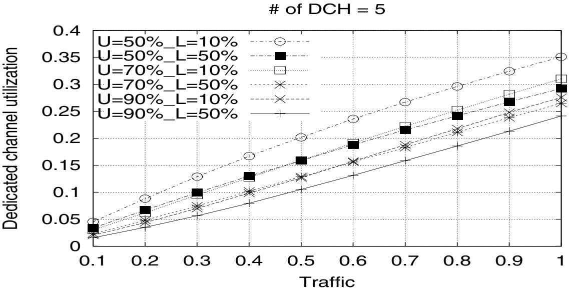


Fig. 7. Dedicated channel utilization (Scheme 2)

If we increase THL while keeping THU at a fixed value, frames in the RLC dedicated buffer will be more frequently transmitted via the common channel and HOL blocking will often occur. Thus, the frame loss probability increases with THL. Although the increase of the dedicated channels can basically contribute to the improvement in the frame loss probability, it is limited due to HOL blocking, as shown in Fig. 4.

Fig. 5 shows the utilization of the dedicated channel as a function of the total amount of traffic at Node B, λ, when the number of dedicated channels is five. When both THU and THL are set to smaller values, the utilization gets larger. The difference in the utilization for different THUs (THLs) is almost insensitive to λ.

Throughout these results, we can say in this scheme that when both THU and THL are set to larger values, frames could be more frequently transmitted on the common channel, whereas the frame loss probability is increasing due to HOL blocking.

4.2 Evaluation of Scheme 2

In this subsection, we focus on the performance of the channel switching Scheme 2. We set the total amount of traffic, λ, to 1.0 and the number of sources, N, to 10.

Fig. 6 shows the frame loss probability in this scheme. Unlike in Scheme 1, as shown in Fig. 4, the frame loss probability is not so sensitive to values of the thresholds used and monotonously decreases with the number of dedicated channels. The reason is that Scheme 2 eliminates HOL blocking due to frames in the MAC-d dedicated buffer for the common channel by allowing use of both dedicate channel and common one at the same time. For example, in order to achieve the loss probability of less than 10^{-5}, we should provide at least five dedicated channels.

Fig. 7 shows the utilization of the dedicated channel when five dedicated channels are available. The characteristics shown there are very similar to that of Scheme 1, as shown in Fig. 5, so that Schemes 1 and 2 do not make a large difference to the utilization.

4.3 Impact of Out-of-Order Transmission

As mentioned in Sec. 2.1, a link layer ARQ protocol is adopted in RNC for transmission error recovery. In this protocol, a receiver will ask the sender to retransmit erroneous frames if necessary. In Scheme 2, frames currently stored in the RLC dedicated buffer are immediately transmitted via the dedicated channel without waiting until all frames stored in the MAC-d dedicated buffer are cleared. Therefore, at the receiver frames transmitted via the common channel may be overtaken by ones transmitted via the dedicated channel, which is referred to as out-of-order transmission since the common channel is shared by several flows, e.g., in a round-robin basis. This will further cause unnecessary retransmission request, resulting in a wasteful use of radio resources. Thus, we will discuss the impact of out-of-order transmission on retransmission property when five dedicated channels are employed. We will define an out-of-order as a case when the frames transmitted over the common channel are overtaken ones over the dedicated channel, and call such overtaken frames out-of-order ones. Furthermore, we define the number of successively overtaken frames as the number of frames transmitted over the dedicated channel until a frame in the MAC-d dedicated buffer is transmitted when out-of-order transmission happened.

Out-of-order Probability. We investigate the out-of-order probability as shown in Fig. 8 in case where $N = 10$ and the number of dedicated channels equals five. We define it as a ratio of the number of out-of-order frames to that of the total received frames at UE. We can see from this figure that the probability increases with THL because the transmission channel more frequently changes from the dedicated one to the common one and vice versa. Fig. 9 shows the probability of the number of successively overtaken frames during an out-of-order transmission. When both THU and THL get smaller, the probability that the number of overtaken frames is two or more becomes smaller. This would result in less retransmission of overtaken frames.

Retransmission Probability for Out-of-order Transmission. We investigate retransmission probability caused by out-of-order transmission. We define the probability, when the acceptable number of out-of-order frames is i, as $p_o \times \sum_{k=i+1}^{\infty} p_{ov}(k)$, where p_o is the out-of-order probability in Fig. 8 and $p_{ov}(k)$ is the probability given by Fig. 9 when the number of successively overtaken frames is k. Fig. 10 shows the retransmission probability. From this result, if receivers can allow five successively overtaken frames, the retransmission probability for out-of-order transmission is less than 10^{-6}. Thus, we can say in this scheme that setting both THU and THL to relatively small value is effective in achieving high throughput.

4.4 Evaluation of Scheme 3

In this subsection, we investigate Scheme 3. We set the total amount of traffic, λ, to 1.0 and the number of sources, N, to 10.

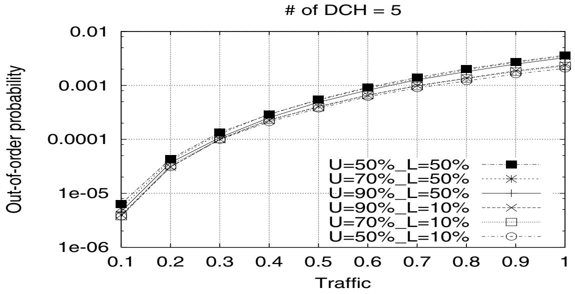


Fig. 8. Out-of-order probability

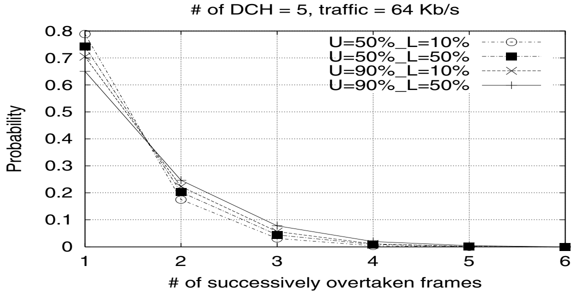


Fig. 9. Probability of the number of successively overtaken frames

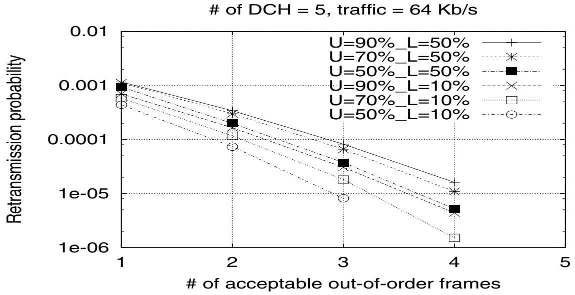


Fig. 10. Probability that frames are retransmitted due to out-of-order arrivals

Fig. 11 shows the frame loss probability. From this result, it is almost the same as that of Scheme 2 as shown in Fig. 6. However, unlike in Scheme 2, the our-of-order transmission never occurs at receivers since frames stored in the RLC dedicated buffer are just transmitted after all frames stored in the corresponding MAC-d dedicated buffer are transmitted via the dedicated channel.

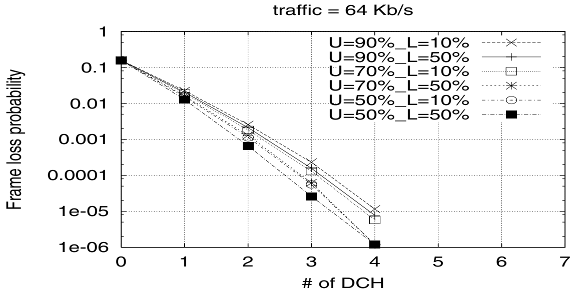
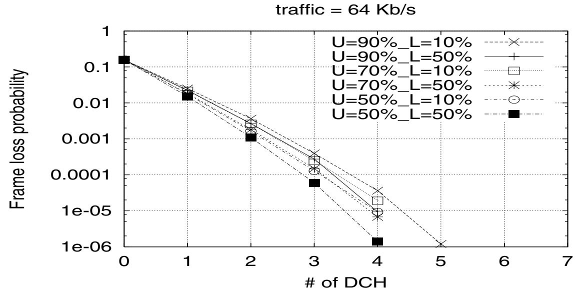
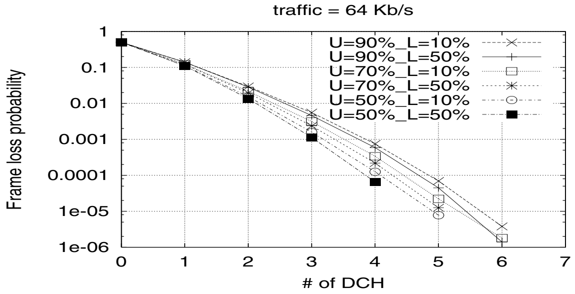


Fig. 11. Frame loss probability (Scheme 3)



(a) Frame loss probability ($N = 20$, $\lambda = 1.0$)



(b) Frame loss probability ($N = 10$, $\lambda = 2.0$)

Fig. 12. Impact of the number of sources and the amount of total traffic (Scheme 3)

4.5 Impact of the Number of Sources and the Amount of Total Traffic

In this subsection, we investigate the impact of the number of sources and the amount of the total traffic in Scheme 3.

Table 1. Switching times per second (Scheme 2)

$L = 10\%$, Scheme 2				
traffic	N	$U = 50\%$	$U = 70\%$	$U = 90\%$
64 Kb/s	10	4.3718	3.3735	2.6458
64 Kb/s	20	4.5017	3.4866	2.7509
128 Kb/s	10	12.9046	10.7725	9.3105

$L = 50\%$, Scheme 2				
traffic	N	$U = 50\%$	$U = 70\%$	$U = 90\%$
64 Kb/s	10	5.0870	3.8251	2.9754
64 Kb/s	20	5.1810	3.9024	3.0665
128 Kb/s	10	17.0085	13.8733	11.6719

Table 2. Switching times per second (Scheme 3)

$L = 10\%$, Scheme 3				
traffic	N	$U = 50\%$	$U = 70\%$	$U = 90\%$
64 Kb/s	10	4.2914	3.2893	2.6013
64 Kb/s	20	4.4010	3.4078	2.7106
128 Kb/s	10	12.5823	10.5501	9.0728

$L = 50\%$, Scheme 3				
traffic	N	$U = 50\%$	$U = 70\%$	$U = 90\%$
64 Kb/s	10	4.984	3.758	2.9343
64 Kb/s	20	5.0627	3.8339	2.9933
128 Kb/s	10	16.3054	13.2505	11.2249

Fig. 12 shows the frame loss probability of Scheme 3 when $N = 20, \lambda = 1.0$ (Fig. 12(a)) and $N = 10, \lambda = 2.0$ (Fig. 12(b)). We can see from Fig. 12(a) that although N increases to 20, the number of required dedicated channels becomes five and it is the same as that in case when $\lambda = 1.0$ and $N = 10$ (See Fig. 11). In addition, if $\lambda = 2.0$ and $N = 10$, the number becomes only six. To further show the effectiveness of Scheme 3, we will indicate the average number of switching times from the common channel to dedicated one per second in Tables 1 (Scheme 2) and 2 (Scheme 3), when the number of dedicated channels is five ($\lambda = 1.0$) or six ($\lambda = 2.0$). These results are related to the channel switching overhead required. It is found that the results in Scheme 3 are smaller than those in Scheme 2 for some THUs and THLs.

5 Concluding Remarks

In this paper, we proposed details of three channel switching schemes according to the specification of RNC in the third-generation mobile telecommunication systems. They are adaptively selecting the appropriate transmission channel from the common channel or the dedicated one in accordance with the queue length of the RLC dedicated buffers. In order to investigate the impact of thresholds in its buffer, we evaluate this performance by simulation. Through numerical results, we have obtained the followings. We first propose Scheme 1 in which the transmission channel switched from the common one to the dedicated one after all frames stored in the corresponding MAC-d dedicated buffer are transmitted via the common channel. Therefore, the frame loss probability of the RLC dedicated buffer cannot be improved by increasing the number of dedicated channels due to HOL blocking effect in MAC-d dedicated buffer.

In Scheme 2, the transmission channel is immediately switched regardless of whether there are any frames stored in the corresponding MAC-d dedicated buffer or not. Thus, the frame loss probability is decreasing with the number of dedicated channels and Scheme 2 is thus more effective than Scheme 1. However, Scheme 2 leads to the out-of-order transmission; i.e., frames via the common channel may be overtaken by ones via the dedicated one, resulting in unnecessary retransmissions. Therefore, a wasteful use of radio resources may occur.

To overcome weak points in Scheme 1 and 2, we also proposed Scheme 3 in which if the transmission of some flows is assigned to the dedicated channels, the corresponding frames in MAC-d dedicated buffer are immediately transmitted via the dedicated channel prior to those in the RLC dedicated buffer. This provides good performance in terms of the frame loss probability as in Scheme 2 without causing the out-of-order transmission as in Scheme 2.

We evaluated the performance of RNC in this paper by focusing only on the case of specific traffic model. Thus, we should also investigate the case where various types of traffic at RNCs for the further work.

Acknowledgments. The authors would like to thank Matsushita Communication Industrial Co., Ltd. for many helpful comments, advice, and suggestions. This work was supported in part by Research for the Future Program of Japan Society for the Promotion of Science under the Project "Integrated Network Architecture for Advanced Multimedia Application Systems" JSPS-RFTF97R16301), Telecommunication Advancement Organization of Japan under the Project "Global Experimental Networks for Information Society Project," and a Grant-in Aid for Scientific Research (12450156) of the Ministry of Education, Culture, Sports, Science and Technology, Japan.

References

1. IMT-2000, *http://www.itu.int/imt/*.
2. R. D. Chrsello, et al, " IMT-2000 Standards: Radio Aspects," *IEEE Personal Communications*, vol. 4, no. 4, pp. 8–40, August 1997.
3. M. Zeng, A. Annamalai, and Vijay K. Bhargava, "Harmonization of global third-generation mobile systems," *IEEE Communications Magazine*, vol. 38, no. 12, pp. 94–104, December 2000.
4. F. Adachi, M. Sawahashi, and H. Suda, "Wideband-CDMA for next-generation mobile communications systems," *IEEE Communications Magazine*, vol. 36, no. 9, pp. 70–80, September 1998.
5. E. Dahlman, B. Gudmundson, Mats Nilsson, and J. Sköld, "UMTS/IMT-2000 based on Wideband CDMA," *IEEE Communications Magazine*, vol. 36, no. 9, pp. 56–69, September 1998.
6. 3rd Generation Partnership Project, *http://www.3gpp.org/*.
7. S. Onoe, K. Ohno, K. Yamagata, and T. Nakamura, "Wideband-CDMA radio control techniques for third-generation mobile communication systems," *Proceedings of IEEE Vehicular Technology Conference*, pp. 835–839, May 1997.
8. 3GPP TS 25.322 V4.0.0 RLC protocol specification, March 2001.
9. 3GPP TS 25.321 V4.0.0 MAC protocol specification, March 2001.
10. VINT Project Network Simulator ns-2, *http://www.isi.edu/nsnam/ns/*.

An Optimal Reservation-Pool Approach for Guaranteeing the Call-Level QoS in Next-Generation Wireless Networks

Fei Hu and Neeraj K. Sharma

Department of Electrical & Computer Engineering, Clarkson University
P. O. Box 5722, Potsdam, New York 13699-5722, USA
{Huf, sharman}@clarkson.edu

Abstract. In order to provide the guaranteed mobile *QoS (Quality-of-service)* for arriving multi-class calls, we need to minimize the dropping rate of handoff calls while at the same time controlling the blocking rate of new calls. This paper proposed a new multi-class call admission control mechanism that is based on dynamically formed reservation pool for handoff requests. The simulation results show that the individual *QoS* criteria of multi-class traffic such as the handoff call dropping probability can be achieved within a targeted objective and the new call blocking probability is constrained to be below a given level. The proposed scheme is applicable to channel allocation of multi-class calls over high-speed multimedia wireless networks.

1 Introduction

Multimedia mobile communications are expected to be the dominant mode of access technology. Besides traditional voice communication, a new range of services such as multimedia, high-speed data, etc. are being offered for delivery over wireless networks. Mobility will be seamless for implementing the blueprint of person's being in contact anywhere and at any time [1-3]. *Mobile Quality-of-Service (M-QoS)* is a set of performance parameters associated with wireless link such as channel error rate and with mobile units such as *Handoff-call Dropping Probability (HDP)* and *New-call Blocking Probability (NBP).* In order to provide higher capacity on the limited radio spectrum, we should use smaller-sized cells such as *pico-* cells instead of *macro-* or *micro-* cells. For such a small cell size, handoff will occur more frequently and make *HDP* a crucial consideration in *M-QoS*. Such handoffs involve allocating sufficient resources in each arriving cell to maintain the *QoS* needs of the established connections. It is a common practice to give a higher priority to the handoff calls as compared to new calls. On the other hand, giving too much priority to handoff calls will result in an excessive *NBP*. Denying of too many new calls can bring an unacceptable ratio of carried-to-admitted traffic and a unsatisfactory revenue for network providers. Various channel allocation schemes have been proposed to implement handoff prioritization and at the same time not hamper the acceptance of new calls.

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 660-671, 2002.

Most of the papers in the literature assume single-class traffic in the cells. The provision of multi-class services (also called multimedia communications) is gaining wide acceptance and will be more ubiquitous in the future wireless and mobile systems.

1.1 Related Works for Multi-class *CAC*

Recently limited work has been reported in the literature regarding CAC schemes in multi-class wireless networks [12-18,20]. In this section we review four different multi-class *CAC* mechanisms which have been proposed in the literature [10,16,18,20].

S. K. Das *et al.* [20] developed an integrated framework for *QoS* provisioning at a lower layer such as the radio link layer combining a novel *CAC* strategy. In this paper we will refer to their scheme as *Low Layer Control Scheme (LLCS)*. *LLCS* can adapt to time-varying and high *Bit Error Rate (BER)* feature of wireless physical link. *LLCS* performs *CAC* on the basis of channel reservation. *QD* in [17] is extensively used in situations where call demands exceed the network's capacity. *LLCS* covers the entire *Network-QoS* which involves multiple layers. Therefore it does not focus on implementation details of channel reservation and handoff prioritization. In our scheme, we adopt the concept of reservation pool for handoff request reservation. This idea is based on increasingly accurate position predicting technology instead of simple *MH* classification and destination determination among three neighboring cells in [20]. This improvement means that we can further reduce the over-reserving of wireless bandwidth. Another *CAC* scheme based on adaptive bandwidth reservation has been proposed by Oliveira et al. in 1998 [16]. We refer it to as *Oliver98* scheme. One of the drawbacks of *Oliver98* strategy is that handoff prioritization, a crucial component of *CAC* mechanism, is based on the concept of *Quality Degradation (QD)*. *QD* should be equally used for all kinds of calls instead of only handoff calls. Another drawback of *Oliver98* strategy is that all of their simulations assume the inter-arrival times of handoff / new calls to follow a geometric distribution, which cannot reflect the actual traffic conditions [18]. The best assumption is general distribution.

Another scheme which we refer to as *Potential Resource Estimation Scheme* (*PRES*) is proposed by Ramanathan in [18]. The obvious drawback of *PRES* is that it shows extremity for handoff prioritization. Handoff prioritization means that we should give handoff calls much higher priority over new calls[1]. However, it does not imply that we should accept all of the handoff calls and consider only the admission control of each arriving new call. If *PRES* is used in practical systems, it may bring unacceptably high *NBP* while minimizing *HDP*. This may lead to network providers' unhappiness due to low revenue resulting from low carried traffic.

One Step Prediction Scheme (*OSPS*) was suggested by Epstein in [10,12-14]. This approach predicts the amount of bandwidth needed in the current cell and each of the

[1] In typical cases, the value of *HDP* is within the range of 10^{-5}~10^{-2}, and the value of NBP is within the range of 10^{-3}~10^{-1}. In other words, *HDP* is generally *100* times larger than *NBP* in the system.

neighboring cells for a specified time interval ahead (called One Step) when a new call of any class arrives. One of the drawbacks of *OSPS* is that it assumes the MH will handoff to all neighboring cells with equal probability when estimating One Step bandwidth. It overestimates the required bandwidth in those neighboring cells and unnecessarily denies many new calls, which makes the *NBP* unacceptably high when *OSPS* is applied to practical *WATM* networks.

1.2 Contributions to Multi-class *CAC* Mechanism

The first contribution to multi-class *CAC* mechanism is that we give a detailed and practical framework for handoff requests reservation. Our discussion assumes an *accurate* next-cell prediction scheme. With the successful application of Kalman filter to *Global Position System (GPS)* and other position locating systems, a precise next-cell prediction technology will become a reality in the next generation mobile networks. It is unnecessary to assume the *MH* will handoff to neighboring cells with undeterminable probability such as in *Oliver98* strategy. It is also incorrect to regard the probabilities to all neighboring cells as the same value such as in *OSPS*. The timing relationship is analyzed between handoff request reservation and later handoff call admission. This is very meaningful for practical system implementation. The state transition map is given for our reservation pool mechanism.

Secondly, for guaranteeing the *M-QoS* of each class of handoff calls, we propose a new notion of *Reservation Ordering (RO)* of handoff requests. *RO* is about the assigning of admission priorities for multi-class calls. However, our admission priority determination is made according to the *MH's* time-varying movement behaviors and the desired *M-QoS* requirements of the multi-class calls themselves. On the other hand, *OSPS* determines call priorities based on only calls' *M-QoS* profiles. For the computation of *RO* value, a weighted algorithm is proposed.

Unlike *LLCS* and *Oliver98* strategy, we assume many traffic classes instead of just two classes (*real-time* and *non-real-time*). The desired amount of bandwidth and delay requirements for these *QoS* profiles can vary greatly. Although *PRES* and *OSPS* also assume multi-class traffic, we analyze urgency details of different *ATM AAL* services instead of simply assuming *K* classes of mobile users. Such urgency details are used for computing *RO* value.

Channel shuffling is our modification of bandwidth compression which is proposed in [20]. Because our channel assigning mechanism involves accurate *MH* identification between handoff request reservation and handoff call admission, we should carry out the shuffling of reservation channels and unoccupied channels at the same time.

Our *CAC* approach is implemented in a distributed way. The algorithm needs only the signaling information between local *BS* and *MH*. This method can bring reduced computation load compared to *MSC*-centered control policy.

Table 1 shows the comparisons between the features of our proposed scheme and those of other four schemes.

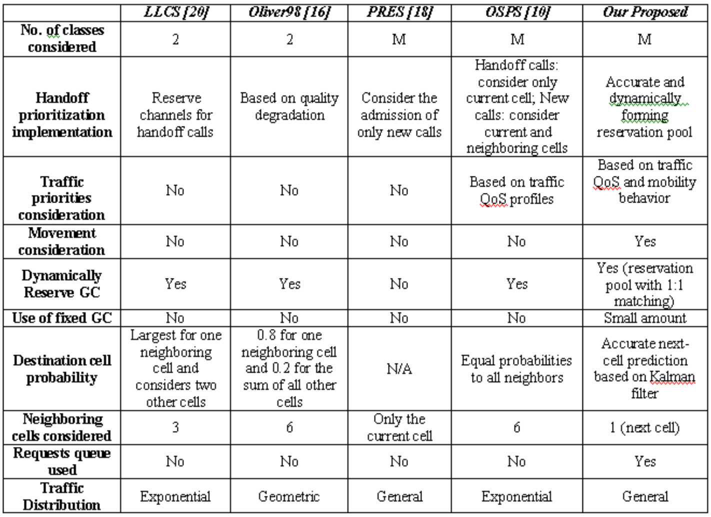
The rest of this paper is organized as follows. Section 2 describes the detailed procedure for forming handoff request reservation pool which is based on accurate next-cell prediction. This is followed by the presentation of *RO* policy in Section 3. Section 4 provides our simulation results and corresponding analysis. Finally, we conclude the paper with a discussion of further work in Section 5.

2 Multi-class Bandwidth Resource Reservation

2.1 Next Cell Prediction

Most of the existing mechanisms for bandwidth reservation and allocation of handoff / new calls assume that we can get the mobility pattern of the *MH* using profile-based schemes. This assumption may not be valid for practical systems. For example, in wireless ATM network environments, wireless components can be connected to *Wide Area Network (WAN), Local Area Network (LAN)* or even Home depending on what kind of ATM network is to be accessed. For such varied wired networks, it may not be possible to predict the arrival of *MH* to some cell since the mobility patterns may not be available. Another drawback for profile-based schemes is that varying traffic conditions suggest that such history-based schemes can never be fully reliable. Therefore we should use real-time position measurements to predict the future path of a moving *MH*. The greatest advantage of future position prediction is that we can determine the next cell which the *MH* will cross with high accuracy. Therefore we need to reserve wireless resources only in next cell among all of the neighboring cells and eliminate the reservation of excessive bandwidth in those neighboring cells where the sum of arriving probabilities is less than some small value. Taking into consideration the limited radio resources compared to wired part of wireless network, such an advantage is valuable. *GPS* can estimate the location of a *MH* with a *95%* probability level within a *100m* margin. However, if differential *GPS* is employed, we can even achieve *3-5m* margin [6].

Table 1. Comparisons of different schemes



	LLCS [20]	*Oliver98 [16]*	*PRES [18]*	*OSPS [10]*	*Our Proposed*
No. of classes considered	2	2	M	M	M
Handoff prioritization implementation	Reserve channels for handoff calls	Based on quality degradation	Consider the admission of only new calls	Handoff calls: consider only current cell; New calls: consider current and neighboring cells	Accurate and dynamically forming reservation pool
Traffic priorities consideration	No	No	No	Based on traffic QoS profiles	Based on traffic QoS and mobility behavior
Movement consideration	No	No	No	No	Yes
Dynamically Reserve GC	Yes	Yes	No	Yes	Yes (reservation pool with 1:1 matching)
Use of fixed GC	No	No	No	No	Small amount
Destination cell probability	Largest for one neighboring cell and considers two other cells	0.8 for one neighboring cell and 0.2 for the sum of all other cells	N/A	Equal probabilities to all neighbors	Accurate next-cell prediction based on Kalman filter
Neighboring cells considered	3	6	Only the current cell	6	1 (next cell)
Requests queue used	No	No	No	No	Yes
Traffic Distribution	Exponential	Geometric	General	Exponential	General

2.2 Determining the Time of Multi-class Handoff Requests Reservation

In this paper, we give a practical way to determine the *Reservation Deadline (RD)* which is a time instance by which bandwidth assignment for the arriving handoff call should be completed. To avoid blind selection of the start point of channel reservation for handoff requests, we define the concept of *Core Area (CA)* with a radius of size *Threshold Distance (TD)* in the current cell as shown in Fig. 1 (Right). In *CA,* there is a high probability for the *MH* to make a dramatic change in its direction and speed. The similar idea is proposed in [6,7]. However, if *MH* moves beyond *CA*, the chances of sudden change of direction are reduced. Thus we can improve the accuracy of next-cell prediction by using Kalman filter. The reasonable position to start making reservations can be chosen as *O* shown in Fig. 1 (Right). From the point of view of *RSS,* position *O* corresponds to the value of *RSS1* in the current cell in Fig. 1 (Left). The relationship between the *RSS* and distance *x* from the transmitter of the *BS* is [19]:

$$RSS_{dB} = -10\gamma \times Log(x) \tag{1}$$

where γ is the propagation path-loss coefficient.

To determine the value of *RD*, we consider the following two criteria:

(1) The *RSS* level of current *BS* drops below a threshold *RSS2* so that it is somewhat difficult to keep the communication with *MH.* The position corresponding to *RSS2* is shown as *A* in Fig. 1 (Right).

(2) The *RSS* level of next-cell *BS* is stronger than that of the current *BS* by a given hysteresis margin Δ. That is, we can only serve handoff calls within the shaded *RSS* range of Fig. 1 (Left).

As can be seen from Fig. 1 (Left), the *RSS* level meeting condition (1) is on the right of line *A*, while for meeting condition (2) is on the right of line *B*. Thus, to meet both conditions, we have to choose right of line *B.* Therefore, once a *MH* arrives at position *B*, we should stop the submitting of handoff request immediately. Then the reservation *Time Duration* Ω for a *MH* is from arriving time at position *O* to arriving time at position *B*. Ω can be expressed as:

$$\Omega = \mathrm{T}_{OB} = t_B - t_O \qquad 2)$$

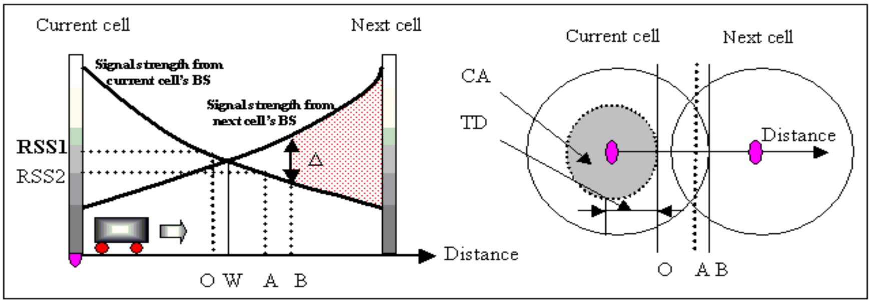


Fig. 1. Time for forming reservation pool (between O and *A*) (Left) *RSS* point of view (Right) Geometry point of view

If we consider predominantly walking and stationary users with an average speed of *2m/s* and a cell radius of *300m*, which is a common case is wireless ATM campus LAN, the typical value of Ω is about *5s ~ 15s* [19]. The value of Ω is important since all of the handoff reservation actions, such as *RO* and overflow request queuing *RO* which will be discussed later, should be finished during Ω. Also the values of *QDT* and *RET* (discussed in Section 5) are setup based on the value of Ω.

2.3 Forming of Multi-class Reservation Pool

Each handoff *MH* sends their bandwidth requirements to the *BS* of next-cell during their own Ω. These handoff request reservations will form a varied-sized pool through marking unoccupied channels from *Free* to *Reserved*. As shown in Fig. 2, handoff calls of different classes can reserve highly varying sized *Channel Blocks (CB)*. The term *CB* comes from the fact that in normal case a handoff call belonging to some class will occupy a series of allocated time slots. The sizes of *Free* and *Occupied* bands are also varying since at any time there are always occupied channels released due to calls completion or handoff to another cell. the dark-shaded channel band is marked as *GC (Guard Channels)*.

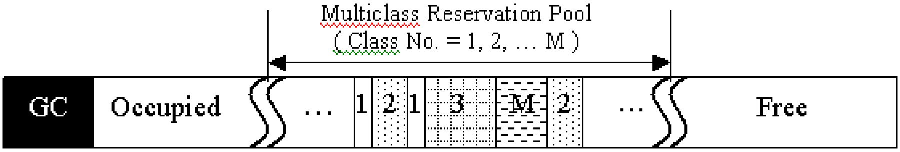


Fig. 2. A snapshot for the channels' status in the current cell

We can draw the *State Transition Map (STM)* as shown in Fig. 3.

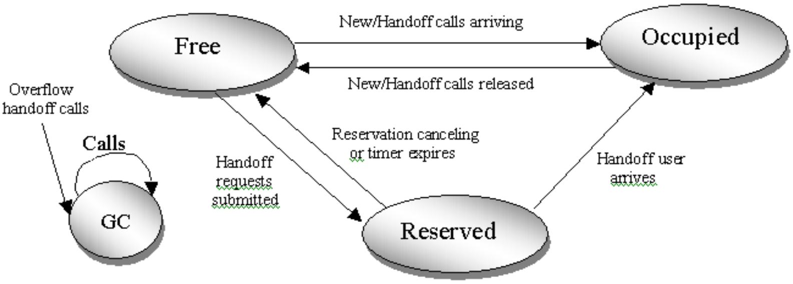


Fig. 3. States Transition Map for the channels with respect to time

3 *Reservation Ordering (RO)* for Multi-class Handoff Calls

The challenging task of bandwidth assignment for multi-class calls is that we should take into consideration largely different *QoS* profiles of each class such as *HDP*, latency tolerance and desired amount of *W-EB*. For multi-class calls, we should assign each class of calls different priorities during resource allocation, unlike in single class case where all calls are assumed to have the same priority. The role of *Reservation Ordering (RO)* is to make sure that the service order for each submitted handoff request reservation is maintained.

For determining the *RO* priority for serving each handoff call, we define a term *Class Urgency (CU)* which represents the desired serving urgency degree. *CU* of the coming multimedia calls is determined by their *M-QoS* parameters such as delay tolerance and *HDP*. However, *CU* cannot be used as the only factor for determining the value of *RO*. For example, when a *MH* is moving almost beyond *reservation area* (from position *O* to position *B* in Fig. 1 (Right)), possibly we should serve this handoff call immediately even though its *CU* is low since its *RSS* from the old *BS* is too weak to continue the communications. In other words, the *RSS* value can become another factor for determining the *RO* priority.

Varying speeds of *MH* can be a serious problem in *WATM* environment where very rapid fading is common due to its small cell size and low used power. To make the situation worse, the *MH* in *reservation area* can wait in traffic jams, traffic lights, or at stop signs. For these cases, it is very improper to assign these *MH* higher priorities just because their *RSS* is low. Since *MH* can travel at different speeds and directions, a faster *MH* will generally require an earlier handoff than a slower one. Thus *MH* velocity can become another important factor for determining the *RO* priority. We can define the *RO* priority as a two-level weighted scheme:

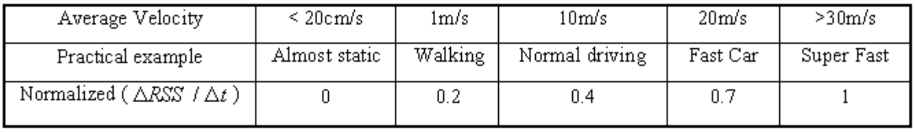
$$RO = \left[W_1 \times (\Delta RSS / \Delta t) + W_2 \times (RSS) + W_3 \times (Class\ \ Urgency)\right] / 3 \qquad (3)$$
$$(W_1 + W_2 + W_3 = 1)$$

where $\Delta RSS / \Delta t$ reflects the value of *MH* velocity, and *RSS* determines the distance of *MH* from its *BS* as shown in formula (3). In multi-class network, we can assign $W_1, W_2, and\ W_3$ based on the significance which above-mentioned three factors may have on *RO*. A reasonable weight suite assignment is:

$$W_1 = 0.1, W_2 = 0.4, \ and \ W_3 = 0.5 \qquad (4)$$

Since *CU* plays such an important role in multimedia network. Note that we should normalize the value of $\Delta RSS / \Delta t$ and *RSS* between *0* and *1*. Table 2 shows a possible velocity normalization.

Table 2. A possible velocity normalization result



Average Velocity	< 20cm/s	1m/s	10m/s	20m/s	>30m/s
Practical example	Almost static	Walking	Normal driving	Fast Car	Super Fast
Normalized (ΔRSS / Δt)	0	0.2	0.4	0.7	1

Note that *RO* depends on two factors. One is *CU* of handoff calls which is only determined by defined *QoS* class. The other is varying mobile behaviors of *MH*. We use velocity ($\Delta RSS / \Delta t$) and position *(RSS)* to symbolize the latter factor. This scheme is different from *OSPS* where calls priorities are only determined by class *QoS* parameters.

There are already many good ways for measuring *MH* velocity such as in [7,8,9,19]. Thus it is not difficult to obtain the value of $\Delta RSS / \Delta t$.

The following pseudo-code describes the necessary system operations each time a *MH* handoff request message is sent to the next-cell's *BS*.

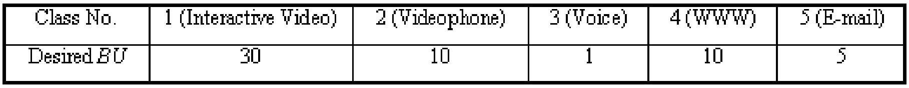
```
Using (7) to compute RO for that MH
   IF this message is a Reservation Canceling
{Re-mark the channels for that MH from 'Reserved' to 'Free' in the pool;
  Delete the buffer unit for that MH in the Reservation Queue if it exists;}
  Else IF this message is a Reservation Confirming_
  IF there is already a 'Reserved' CB for that MH in the pool
      {Modify its RO to the new value;
  Reorder all the CB based on their new RO value in the pool;}
      Else    /* This is a new reservation */
  {Delete the buffer unit for that MH in the Reservation Queue if it exists;
   IF available free bandwidth $\geq$ Desired bandwidth
   {Insert a new CB in the reservation pool based on RO priority}
  Else    /* available free bandwidth < Desired bandwidth */
  {Buffer it into the Reservation Queue}♣
```

4 Simulation Experiments

4.1 Simulation Model

Based on the proposed *CAC* algorithm we built a C-based simulator. In this simulation we choose the total capacity of the current cell as *10,000*[3] *Bandwidth Units (BU).* The *BU* requirements for the five classes of calls are chosen as shown in Table 3.

Table 3. *BU* requirements for the five classes



Class No.	1 (Interactive Video)	2 (Videophone)	3 (Voice)	4 (WWW)	5 (E-mail)
Desired *BU*	30	10	1	10	5

The cell radius is assumed to be *500m,* which is a typical size for future *WATM* system. Three different velocities are assumed: *2m/s* (walking), *10m/s* (normal-speed car), and *20m/s* (high-speed vehicle). Furthermore we assume that the five classes of

[3] In this simulation, we choose this capacity value only for testifying the effect of our scheme. As a matter of fact, future *WATM* or even *IMT-2000* should be expected to be able to provide an aggregate transmission capacity of *25 Mb/s* when such systems are offered at frequency bands above *3 GHz.*

calls have the same percentages of three velocities in order to emphasize the influence of *class urgency* on the computation of *RO*. A cluster of seven cells is assumed and each cell keeps contact with its six neighboring cells.

4.2 The Role of Queue

Our approach uses a queue for storing overflowing handoff reservations due to the lack of *free* channels. To investigate the effect of the queue, we assume the same numbers of five class of handoff requests, that is, their percentage within the total handoff requests is *20%* individually. Because handoff congestion happens only when *HTL* is high, we let *HTL = 80%,* which makes the *HDP* almost ten times larger than the *HTL = 50%* case.

The *HDP* results of five classes of handoff calls are shown in Fig. 4. Although each class of handoff calls experience a certain degree of improvement for their *HDP* due to the introduction of reservation queue, the improvement values are different. It can be seen that *class 5* calls have the most dominant decreasing *HDP* while *class 1* calls have the least improvement compared to *no queue* case. A possible explanation for this phenomenon is that *class 5* calls have the lowest serving priority among the five classes of calls since only *Class Urgency* is crucial for computing the value of *RO* after the elimination of other factors such as mobile movements. Since the percentage of *class 5* users is the same as other classes, *class 5* calls will have the largest probability for being buffered into reservation queue. Therefore they benefit the most from reservation queue.

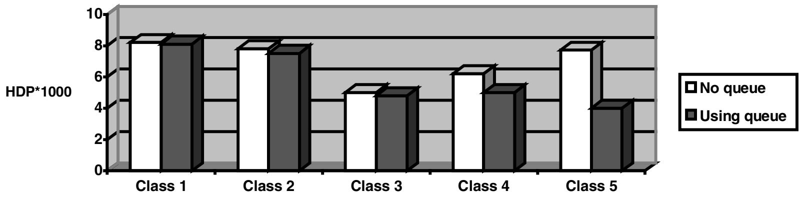


Fig. 4. The importance of reservation queue

4.3 The Importance of Determining Multimedia Servicing Prioritization

If we assume that *MH's* position and velocity cannot influence much on the *RO* of each handoff call except for the *CU* of each class[4], we can see the effect of *RO* on improving *HDP* of each class of handoff calls.

We only consider two classes of calls: *class 1* and *class 5*, since *class 1* calls have the most crucial urgency requirements while *class 5* calls have the least urgency requirements. Two important cases are considered: light handoff load *(HTL = 25%)* and

[4] This can be done by assuming each class of calls have the same percentage of all types of moving users such as pedestrians and cars.

heavy handoff load *(HTL = 75%).* The reason for choosing these two extreme cases is that we may see the effect of *RO* on *HDP* more clearly.

Figure 5 (a) ~ (d) are our simulation results. The *X-axis* represents the percentage of a given class of calls among all handoff calls. It varies from *20% to 100%.* The *Y-axis* is the value of *HDP* multiplied by *10,000.* It can be seen that *HDP* of *Class 1* calls decreases when *RO* is adopted. Although in light handoff load case, the reduction is not very obvious (Fig. 5 (a)), in heavy handoff load case the effect of *RO* is very dominant (Fig. 5 (b)). This is not a surprising result since *RO* can assign *class 1* calls the highest priority when only *CU* is considered.

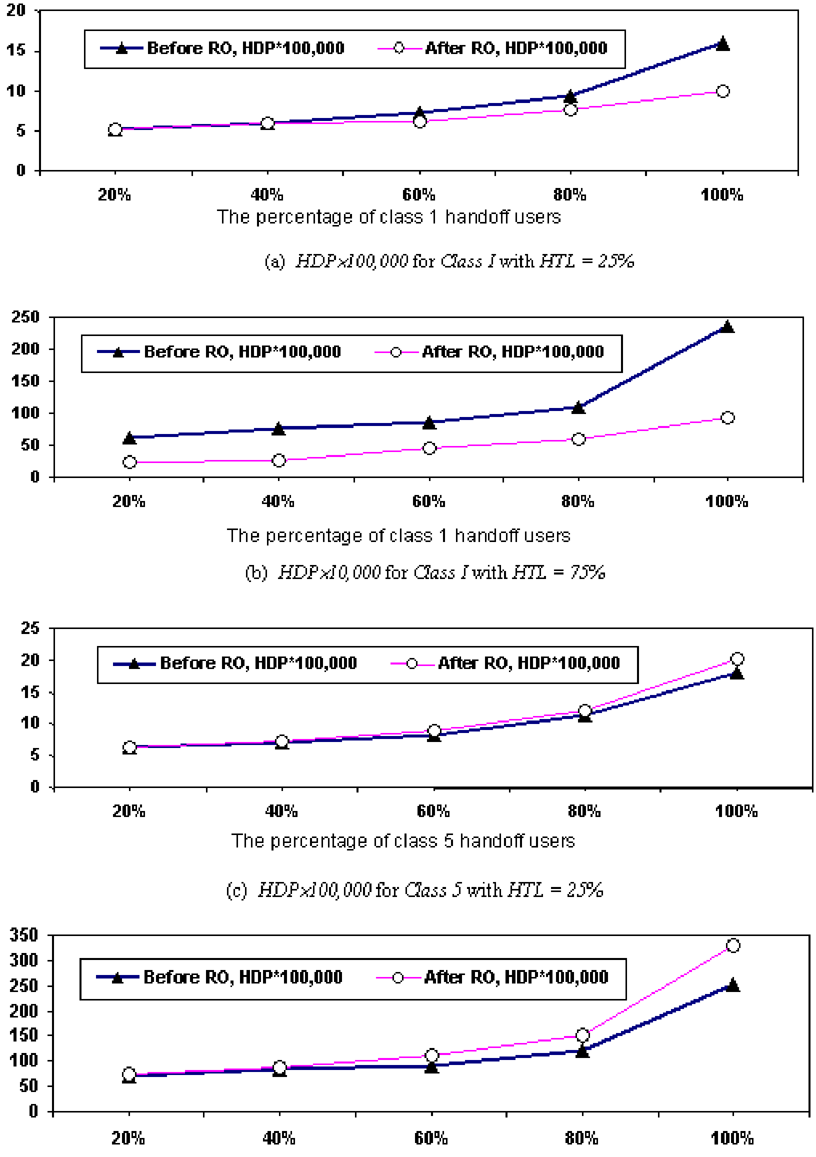
Unfortunately, *HDP* increases for *class 5* calls (Fig. 5 (c) and (d)), especially in heavy handoff load case (Fig. 5 (d)). This is because *class 5* calls get the lowest priority when their *RO* is compared to other classes. When the network is under congestion, the *class 5* calls have the highest probability for being dropped among the five classes.

For dealing with this problem, we can use the crossover ATM switch to buffer those delay-insensitive *class 5* ATM cells. When the handoff connection is rerouted from the old path to a new path, a crossover switch should be found out using fast searching algorithm [5]. Thus, the down-link data stream can be stored in the buffer of this switch.

5 Conclusions and Future Work

This paper addressed the problem of providing *M-QoS* guarantee for multi-class calls in the *WATM* network. The network is assumed to be able to accurately predict next-cell which the *MH* will cross. This assumption is reasonable for the developing mobile position system such as *GPS.* A multi-weighted algorithm for computing priorities of handoff requests was proposed in order to serve arriving multi-class calls with highly diverse *QoS* parameters. A dominant feature of our approach is combining practical handoff behaviors with the call admission procedure. This includes the *RO* computation and the notion of three timers. Several important considerations for practical system implementation were discussed in this paper.

In the introduction of this paper we mentioned that we focus on the *LCA* mechanism instead of *CCA* mechanism. However, there is close relationship between these two mechanisms. A typical example is *Channel Borrowing Mechanism (CBM)* [4]. *CBM* states that the whole capacity of any cell is not a fixed value. Each cell only keeps a set of nominal channels (less than *FCA* case) and can borrow free channels from its neighboring cells to accommodate new calls. Thus, the *NBP* can be further decreased. One of our future tasks is combining the *CBM* with our proposed approach to investigate the improvement of *NBP.* Another future task is to derive analytical models to evaluate the performance of our *CAC* scheme. As shown in Fig. 1, this paper provides a reservation-based call admission strategy for guaranteeing the network *QoS*. Further work in this area will include translating the high-level resource allocations into scheduling at the low levels such as *MAC* layer so as to map the network *QoS* to *MAC-oriented QoS.*



(a) *HDP×100,000* for *Class 1* with *HTL = 25%*

(b) *HDP×10,000* for *Class 1* with *HTL = 75%*

(c) *HDP×100,000* for *Class 5* with *HTL = 25%*

(d) *HDP×100,000* for *Class 5* with *HTL = 75%*

Fig. 5. The influence of *RO* on *HDP*

References

1. Mansoor Shafi, *et al.*, "Wireless Communications in the Twenty-First Century: A Perspective," *Proceedings of the IEEE*, vol. 85, no. 10, pp. 1622-1637, October 1997.

2. Upkar Varshney and Ron Vetter, “Emerging Mobile and Wireless Networks,” *Communications of the ACM*, vol. 43, no. 6, pp. 73-81, June 2000.
3. Lajos Hanzo, “Bandwidth-Efficient Wireless Multimedia Communications,” *Proceedings of the IEEE*, vol. 86, no. 7, pp. 1342-1380, July 1998.
4. I. Katzela and M. Naghshineh, “Channel Assignment Schemes for Cellular Mobile Telecommunication Systems: A comprehensive Survey,” *IEEE Personal Communications*, pp. 10-31, June 1996.
5. C-K Toh, “Wireless ATM and Ad-Hoc Networks : Protocols and Architectures,” pp. 88-98, Kluwer Academic Publishers, 1997 (*ISBN: 079239822X*).
6. Ming-Hsing Chiu and Mostafa A. Bassiouni, “Predictive Schemes for Handoff Prioritization in Cellular Networks Based on Mobile Positioning,” *IEEE J. Select. Areas Commun.*, vol. 18, no. 3, March 2000.
7. Tong Liu, *et al.* “Mobility Modeling, Location Tracking, and Trajectory Prediction in Wireless ATM Networks,” *IEEE J. Select. Areas Commun.*, vol. 15, pp. 1208-1225, Sept. 1997.
8. Tong Liu , “An Optimal Self-Learning Estimator for Predicting Inter-Cell User Trajectory in Wireless Radio Networks,” *IEEE Globecom,* 1997.
9. Martin Hellebrandt, *et al.,* “Estimating Position and Velocity of Mobiles in a Cellular Radio Network,” *IEEE Trans. on Vehicular Technology*, vol. 46, no. 1, Feb. 1997.
10. B.Epstain and M.Schwartz, “Reservation strategies for multimedia traffic in a wireless environment,” *Proc. 1995 IEEE 45th Vehicular Technology Conf.*, pp. 165-169, July 1995.
11. Larry L. Peterson and Bruce S. Davie, “Computer Networks: A System Approach,” Morgan Kaufmann Publishers, San Francisco, California, pp. 649-662, 2000. (*ISBN: 1558605142*)
12. Bracha M. Epstein and Mischa Schwarz, “Predictive QoS-Based Admission Control for Multiclass Traffic in Cellular Wireless Networks,” *IEEE JSAC*, vol. 18, no. 3, pp. 523-534, March 2000.
13. Bracha M. Epstein and Mischa Schwarz , “QoS-based predictive admission control for multi-media traffic,” in *Broadband Wireless Communications*, M. Luise and S. Pupolin, Eds. Berlin, Germany: Springer-Verlag, pp. 213-224, 1998.
14. Bracha M. Epstein, “Resource Allocation Algorithms for Multi-Class Wireless Networks,” Ph.D. dissertation, Columbia Univ., New York, 1999.
15. C. Chao and W. Chen, “Connection Admission Control for Mobile Multi-Class Personal Communications Networks,” *IEEE JSAC*, vol. 15, no. 8, pp. 1618-1626, October 1997.
16. Carlos Oliveira, *et al.,* “An Adaptive Bandwidth Reservation Scheme for High-Speed Multimedia Wireless Networks,” *IEEE JSAC*, vol. 16, no. 6, pp. 858-873, August 1998.
17. K. Seal and S. Singh, “Loss profiles: A quality of service measure in mobile computing,” *Wireless Networks*, vol. 2, no. 1 , January 1996.
18. Parameswaran Ramanathan, *et al.,* “Dynamic Resource Allocation Schemes During Handoff for Mobile Multimedia Wireless Networks,” *IEEE JSAC*, vol. 17, no. 7, pp. 1270-1283, July 1999.
19. Howard G. Ebersman and Ozan K. Tonguz, “Handoff Ordering Using Signal Prediction Priority Queuing in Personal Communication Systems,” *IEEE Trans. on Vehicular Technology*, vol. 48, no. 1, pp. 20-35, Jan. 1999.
20. S. K. Das, *et al.,* “A Call Admission and Control Scheme for Quality-of-Service (QoS) Provisioning in Next Generation Wireless Networks,” *Wireless Networks* 6, pp. 17-30, 2000.

A New Adaptive Channel Reservation Scheme for Handoff Calls in Wireless Cellular Networks

Zhong Xu[1], Zhenqiang Ye[1], Srikanth V. Krishnamurthy[2], Satish K. Tripathi[2], and Mart Molle[2]

[1] Department of Electrical Engineering
[2] Department of Computer Science and Engineering,
University of California at Riverside, Riverside, CA92521, USA

Abstract. In wireless cellular networks, in order to ensure that ongoing calls are not dropped while the owner mobile stations roam among cells, handoff calls may be admitted with a higher priority as compared with new calls. Since the wireless bandwidth is scarce and therefore precious, efficient schemes which allow a high utilization of the wireless channel, while at the same time guarantee the QoS of handoff calls are needed. In this paper, we propose a new scheme that uses GPS measurements to determine when channel reservations are to be made. It works by sending channel reservation request for a possible handoff call to a neighboring cell not only based on the position and orientation of that call's mobile station, but also depends upon the relative motion of the mobile station with respect to that target cell. The scheme integrates threshold time and various features of prior schemes to minimize the effect of false reservations and to improve the channel utilization of the cellular system. Simulation results show that our scheme performs better in almost all typical scenarios than prior schemes.

Keywords: cellular networks, handoff, adaptive channel reservation

1 Introduction

As a mobile station (MS)[1] moves from one cell to another, its ongoing call is handed-off from the old cell to a new cell. This requires that the call be accommodated by the new cell. Since dropping a handoff call is more annoying than blocking a new call from user's perspective, handoff calls should be given higher priority than new calls. It has been shown that the method by which handoff is achieved has a significant impact on the network's performance [1]. Due to the inherent bandwidth limitation in wireless cellular networks, micro/pico cellular architectures are attractive for achieving higher system capacity [2]. In this case, the coverage area of a cell will be defined by a circular region that is a few hundred meters to a few kilometers in radius. As a direct result, the rate of handoffs increases dramatically even when MSs move at low speed.

[1] We use "MS" to represent "one MS with an ongoing call" in the rest of this paper.

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 672–684, 2002.

The probability of an ongoing call being dropped due to a handoff failure and the probability of a new call being blocked due to the temporary unavailability of an idle channel are major metrics that define the performance of cellular systems. The handoff prioritization schemes implemented in the network have a significant impact on these two probabilities. All the handoff prioritization schemes have a common characteristic: ensuring a lower handoff dropping probability at the expense of an increased new call blocking probability. Efficient handoff prioritization schemes are those allow a high utilization of the wireless bandwidth (by accommodating a higher number of new calls) while guarantee the QoS of handoff calls.

The naive channel assignment strategy is to treat handoff calls and new calls equally [3]. This scheme would result in the new call blocking probability and the handoff call dropping probability being equal. Obviously, this scheme performs poorly when the offered load on the network is high. Much work has been done on handoff prioritization in wireless cellular systems [3,5,6,7,8]. Basically there are two strategies that are popular for prioritizing handoff calls [2]: the *guarded channel strategy* and the *handoff queueing strategy*. The guarded channel strategy decreases the handoff dropping probability by reserving a fixed number of channels exclusively for handoff calls. New calls will be blocked if the number of idle channels is equal to or less than the number of guarded channels, while handoff calls can be served until all the channels are occupied. The handoff queueing strategy is a way of delaying handoff due to the temporary unavailability of channels. The mobile switching center (MSC) queues the handoff requests instead of denying access if the candidate cell has no idle channel available. Queueing is possible due to the overlapping region between adjacent cells where it can communicate with both the old and the new base station (BS). The maximum queueing time is limited by the MS' dwell time in the overlapping area. If the traffic load is heavy, or if the maximum allowed queueing time is very small, it is highly unlikely that a queued handoff request will be entertained. These two strategies can be combined to obtain better performance as compared with the individual strategies [7].

Since the mobility behavior of different MSs may be totally different, and the traffic load offered in each cell varies from time to time, any static channel reservation scheme cannot work efficiently all the time. In order to solve this problem, several adaptive (dynamic) channel reservation schemes have been proposed [4, 5,6,7]. The *shadow cluster* concept proposed in [4] allows the base station of each cell to calculate the probabilities that a MS will be active in other cells at future times, and thereby facilitate the prediction of future resources demands. In [6], the number of guarded channels in each cell is adjusted according to the current estimate of the handoff call arrival rate, which is derived from the current number of ongoing calls in neighboring cells and the mobility patterns of the MSs. In [5], channels are dynamically reserved by using the *request probability* determined by the mobility patterns of the MSs and the current traffic load. All these schemes take into account the MSs' mobility patterns when they dynamically make channel reservations. But the mobility patterns that are considered are all

MSs' general patterns, and they do not identify each individual MS's mobility behavior separately. In [7], the Predictive Channel Reservation (PCR) scheme is proposed and is based on mobile positioning. The *threshold distance* concept (See II.A for its definition) is used to define the size of channel reservation area. The PCR scheme makes predictive channel reservations for each MS based on its current position and orientation. But the threshold distance in the PCR scheme is constant for all MSs.

In this paper, we propose a new handoff prioritization scheme, which is called adaptive channel reservation (ACR) scheme. The ACR scheme integrates the features of threshold time, reservation queueing, reservation cancellation and reservation pooling to minimize the false reservations and to improve the channel utilization of the cellular system. Like [7] , the ACR scheme is also based on GPS measurements [9]. We don't discuss GPS further in this paper, and just make an assumption that each MS is equipped with GPS and can obtain its position information in real-time.

The remainder of this paper is organized as follows. Section 2 outlines the ACR scheme. In section 3, we describe the models that we use for simulating the ACR scheme. Detailed performance results are presented and interpreted in section 4. Finally, we present our conclusions in section 5.

2 Adaptive Channel Reservation

In the ACR scheme, channel reservation decisions are made based not only on each MS' current position and orientation, but also on the *relative* moving speed with respect to its next target cell. Each MS [2] measures its coordinates at regular time interval (every ΔT seconds) using GPS. The coordinate information is piggybacked onto uplink data packets (or sent to the associated BS by means of special uplink packets). The BS keeps track of each MS' previous positions, predicts its trajectory [13] and calculates the relative moving speed with respect to the next cell that the MS is predicted to enter. Based on these calculations, we can predict the time within which the MS will reach this candidate cell.

In [7], the *threshold distance* (D_{th}) is defined as the radius of a circle which is co-centered with a cell, and this circle is smaller than the cell's coverage area (Figure 1(a)). The area between these two circles is called the channel reservation area. When a MS enters the reservation area of a cell from the inner part of that cell (or a new call is generated inside the reservation area), and at the same time, is heading to a new cell, a reservation request will be sent to that new cell's BS. There are some problems in the PCR scheme. Suppose a MS moves into the reservation area of cell A and heads to cell C (See Figure 1(a)); although the MS is located in the reservation area of cell A, there is a long distance between the MS's current location and the boundary of cell C. After the MS reserves a channel, it may need a long time to move into cell C. In this case, the time for which a channel is reserved for this MS will be too long, and thus the overall channel utilization will deteriorate. Another problem is that each MS has its own

[2] We assume that every MS can carry at most one call at a time in this paper.

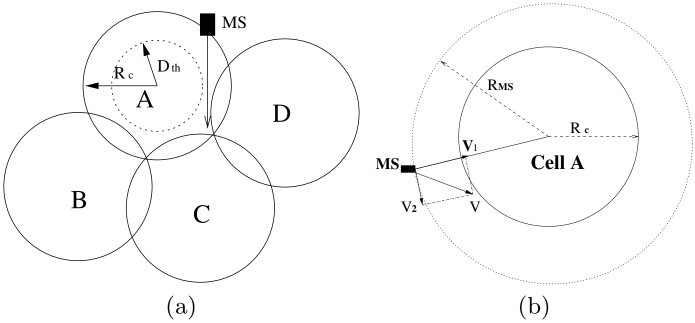


Fig. 1. Threshold distance in the PCR scheme (a) and threshold time in the ACR scheme (b)

motion pattern and hence it is inappropriate to define one constant threshold distance for all MSs. One extreme example is that there is a MS located in the overlapping area of two adjacent cells, the moving speed of this MS is very slow (close to 0). If the PCR scheme is used, two channels (each cell has one channel occupied) will be occupied by this call, one channel is used for communication in the current cell and the other is reserved for this call in the adjacent cell. Since the MS of this call is almost stationary, the reserved channel may not be used for the life time of this call. Naturally, this method leads to the under-utilization of wireless channels.

2.1 Threshold Time

In order to solve these problems, we use *threshold time (T_{th})* instead of threshold distance to reflect possible reservation requests. Here T_{th} is a constant time value. According to each MS' current moving speed, orientation and location information, BSs can predict the time within which the MS will reach the boundary of its next target cell. In Figure 1(b), a MS is moving with a velocity V towards cell A. The velocity V can be decomposed into two orthogonal component vectors, V_1 and V_2, where V_1 is the velocity component of this MS towards the center of cell A. From V_1 and R_{MS} (the distance between the MS and the center of cell A), we can estimate the time T_p by which the MS will reach the boundary of cell A.

$$T_p = \frac{R_{MS} - R_c}{V_1} \quad . \tag{1}$$

where R_c is the radius of cell A.

If $T_p > T_{th}$, it means that the MS may take a time longer than T_{th} to reach cell A, and it does not need a channel reservation in that cell at current time. If $T_p \leq T_{th}$, it means that the MS under consideration will move into cell A soon, and a reservation request will be sent by the current BS to cell A's BS. Suppose $R_{th} = R_{MS}|_{T_p=T_{th}}$; we call R_{th} the threshold distance for this MS. Note that the threshold distance defined in our paper is different from that in [7] in that different MSs have different threshold distances even though all the MSs have the same T_{th}, because they have different relative moving speeds.

Like the PCR scheme, in our scheme, threshold time is integrated with reservation queueing, reservation cancellation and reservation pooling to minimize the effect of false reservations and to improve the channel utilization of cellular systems. In the following paragraphs, we briefly describe the concepts of reservation cancellation, reservation pooling and reservation requests queueing defined in [7].

2.2 Reservation Requests Queueing

If a reservation request is received by the BS of one cell, and there is no idle channel available, this reservation request will be put into a reservation queue. If the reservation queue is not empty, a channel released by a call (either complete or handed-off to a neighboring cell) is added to the reservation pool at once and one reservation request is dequeued.

2.3 Reservation Cancellation

A reservation may be invalid (false reservation) at a later time because the MS may change its moving direction, slow down its moving speed or because the call may terminate before the MS reaches the candidate cell. In this case, the false reservation will be canceled and a reserved channel will be released (if the reservation queue is empty) or one reservation request is deleted from the reservation queue (if the reservation queue is not empty). The frequency of occurrence of false reservations depends primarily on the MSs' mobility pattern and prediction accuracy of future movement.

2.4 Reservation Pooling

Rather than strictly mapping each reserved channel to the MS that made the reservation, the set of reserved channels, at any moment, is used as a generic pool to serve handoff requests. Once a handoff is needed, the BS will randomly choose a reserved channel from the reservation pool and assign it to the handoff call. So when one BS sends a reservation request to another BS, it does not need to send the MS' ID.

The overhead incurred by the ACR scheme is the prediction of each MS' future trajectory, the transmission of reservation requests and reservation cancellation messages among BSs. Because all of these functions can be performed by BSs, there is no extra overhead for MSs (for which computation power is limited). The communication overhead (among the BSs) is transmitted over wire-line network, and does not consume the precious wireless bandwidth.

3 Simulation Model

We construct a simulation model to evaluate the performance of the ACR scheme. The model is implemented in CSIM18 [14]. This simulation model includes system topology model (cell model), traffic model and user mobility model.

3.1 Cell Model

The simulation is conducted over a L layer micro cellular mobile radio system (See Figure 2). Square, circular or hexagonal cells are commonly used in the simulation of wireless cellular systems. In our simulation we use hexagons to represent the neighborhood relationships among cells and circles to approximate the coverage area of cells. There are overlapping areas between adjacent cells. The radius of each circle (or hexagon) is represented by R_c. There is one central cell in the topology (first layer). The central cell is surrounded by six cells which make up the second layer. There are 12 cells in the third layer, and $6(i-1)$ cells in the ith layer $(1 < i \leq L)$.

In order to avoid border effects, when a MS moves out of the service area of the system, this MS will be wrapped around to re-enter the system from the other side. Such a toroidal arrangement is an efficient way to approximately simulate very large cellular systems.

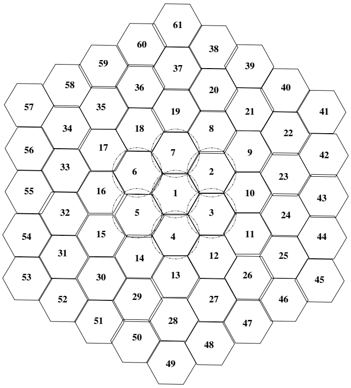


Fig. 2. The topology of a 5-layer wireless cellular system

3.2 Traffic Model

In our simulation model we only consider homogeneous calls, and assume that each MS needs only one channel per call. Call generation in the system follows a

Poisson process with an average arrival rate λ. The call holding time T_c follows an exponential distribution with an average service rate μ_c. The number of channels in each cell is a constant c. The normalized offered traffic load of the system is defined to be

$$\frac{\lambda}{\mu_c \cdot N \cdot c} \ , \tag{2}$$

where N is the number of cells in the system, and is given by:

$$N = 1 + \sum_{i=2}^{L} 6(i-1) = 3L(L-1) + 1 \ \cdot \tag{3}$$

Note that the load is measured in Erlang.

3.3 Mobility Model

Several mobility models, such as the random-walk model and the fluid-flow model are often used to depict MS' moving behavior in simulations and analyses [10,11, 12]. In our simulation, we consider a more realistic mobility model. When a new call is generated, the MS initially chooses a speed which is uniformly distributed over $[V_{min}, V_{max}]$ and a moving direction which is uniformly distributed over $[0, 360°)$. In each variable-length period T_u (which is exponentially distributed with mean $E[T_c]/3$), the MS moves along a straight line. After that period, the MS may stop (with a probability P_{stop}) for a time T_u or continue to move (with a probability $P_{cont.} = 1 - P_{stop}$). If the MS continues to move, it may change its moving direction. The MS makes ±90° turns with probability $P_{90°}$, makes ±45° turns with probability $P_{45°}$, and moves along the original moving direction with probability $P_{0°} = 1 - (P_{90°} + P_{45°})$.

In order to evaluate the effects of speed patterns on the system performance, three different speed patterns are defined.

- **V1:** $V_{min} = 0$ and $V_{max} = 20m/s$.
- **V2:** $V_{min} = 0$ and $V_{max} = 5m/s$.
- **V3:** $V_{min} = 15m/s$ and $V_{max} = 20m/s$.

Compared to V1, V2 is low speed moving pattern, and V3 is high speed moving pattern.

4 Performance Evaluation

We define the new call blocking probability P_b, the handoff dropping probability P_d and the call incompletion probability P_{nc} as the system performance metrics. Call incompletion probability is the probability that a call is not completed (either due to being blocked or because of being dropped during handoff). The values of the various parameters used in simulation are listed in Table 1. In order to evaluate the performance of the ACR scheme, we simulate the PCR scheme with the same simulation model and under the same system conditions, and compare its performance results with that of the ACR scheme.

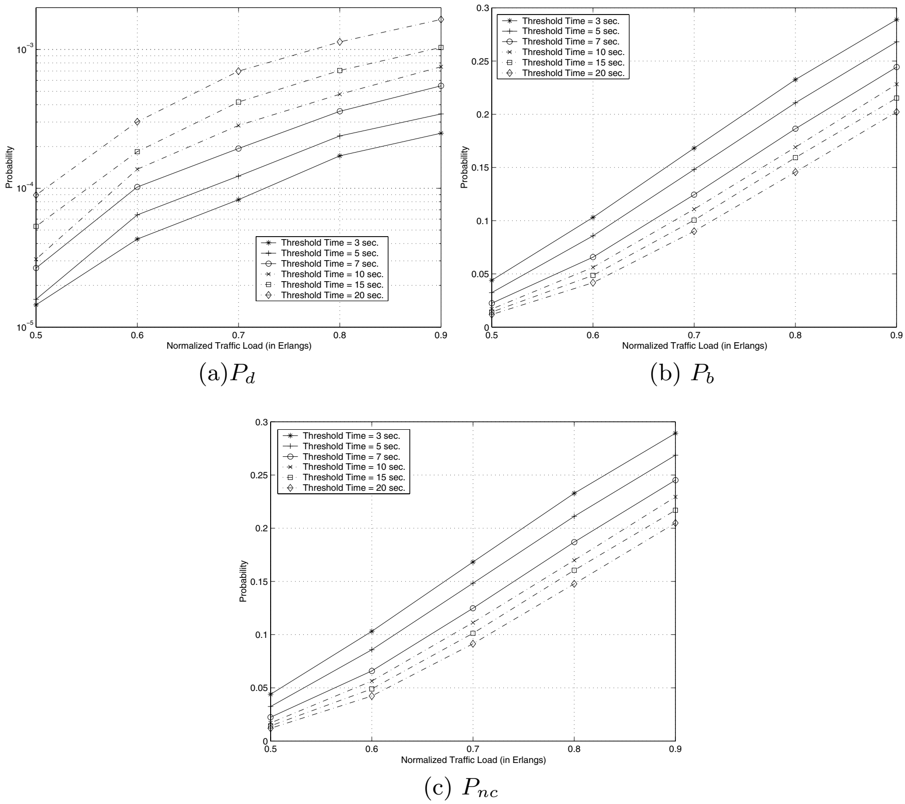
Table 1. Parameter Values in the Simulation

Parameter	Value	Description
L	5	Cell Layer Number
R_c	500m	Cell Radius
c	20	Number of Channels in each Cell
T_c	180s	Call Holding Time
P_{stop}	0.1	Probability with which a MS stops
$P_{cont.}$	0.9	Probability with which a MS continues motion
P_{0°	0.7	Probability of Keeping Original Moving Direction
P_{45°	0.1	Probability of Making a 45° turn
P_{90°	0.2	Probability of Making a 90° turn

Figure 3 shows the P_b, P_d and P_{nc} experienced by the system when the ACR scheme is used with different values of T_{th}. The MSs move in accordance to the speed pattern V1. From Figure 3(a) it is seen that P_d decreases from 0.17% to 0.025% with the increase of T_{th} (from 3 seconds to 20 seconds) when normalized traffic load is 0.9; the penalty incurred is that, P_b increases from 20% to 29% (See Figure 3(b)). We also find that P_d is already very small (compared to P_b) even when T_{th} is very small (for example, $T_{th} = 3$ seconds). The reason for this result is that the overlapping area of a cell with its neighboring cells contributes a fairly large portion (about 35%) of the entire cell, and even if a MS can not access an idle channel before it traverses the boundary of its next cell, it still has a certain period of time (its dwell time in the overlapping area) to wait for an idle channel. So its maximum allowed channel waiting time is larger than T_{th}. Since P_d is much smaller than P_b, most of the unsuccessful calls are new calls (which are blocked), therefore the call incompletion probability P_{nc} and the new call blocking P_b are almost the same (See Figure 3(c)).

Figures 4 and 5 show the performance of the ACR scheme under two different speed patterns (low speed pattern V2 and high speed pattern V3). From these two figures, it is seen that P_d under the low speed pattern is a little higher than that under the high speed pattern. On the other hand, P_b is lower under the low speed pattern. Under the low speed pattern, the possibility that an ongoing call is handed-off to another cell is smaller than that under the high speed pattern. As a result, under the low speed pattern, the number of handoffs is smaller, and consequently, the number of reserved channels at any given time is also smaller. Since the ACR scheme is incorporated with reservation pooling, the more the number of reserved channels, the lower the probability P_d. This is the reason for P_d being high and P_b being low under the low speed pattern. Another observation is that both P_d and P_b are not very sensitive to a change in T_{th} under the low speed pattern. On the other hand, the sensitivity is much higher under the high speed pattern.

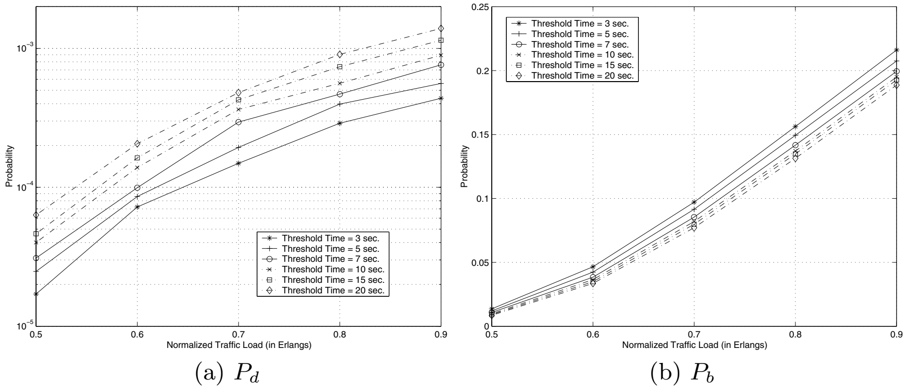
The comparison of the ACR scheme with the PCR scheme in terms of performance under the speed pattern V1 is shown in Figure 6. By choosing $T_{th} = 3$ seconds in the ACR scheme and the threshold distance $D_{th} = 0.723R_c$ in the PCR scheme, the two schemes have almost the same P_d for various normalized traffic loads (See Figure 6(a)). The ACR scheme decreases P_b by 4.5% as compared with the PCR scheme (See Figure 6(b)). In other words, for a given



(a)P_d (b) P_b

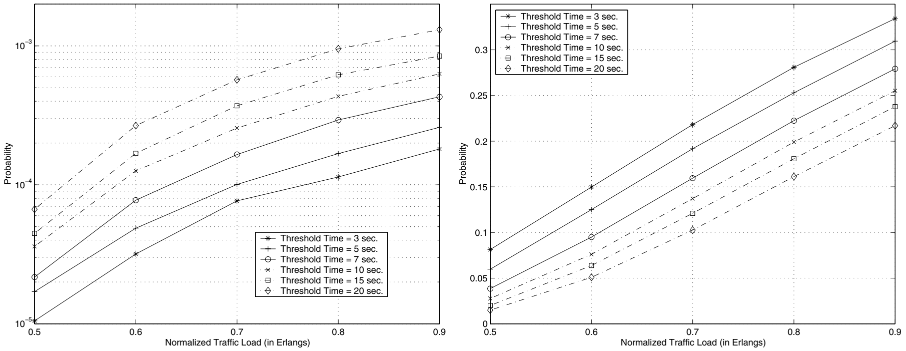
(c) P_{nc}

Fig. 3. Performance of the ACR scheme under speed pattern V1



(a) P_d (b) P_b

Fig. 4. Performance of the ACR scheme under speed pattern V2 (low speed pattern)



(a)P_d (b)P_b

Fig. 5. Performance of the ACR scheme under speed pattern V3 (high speed pattern)

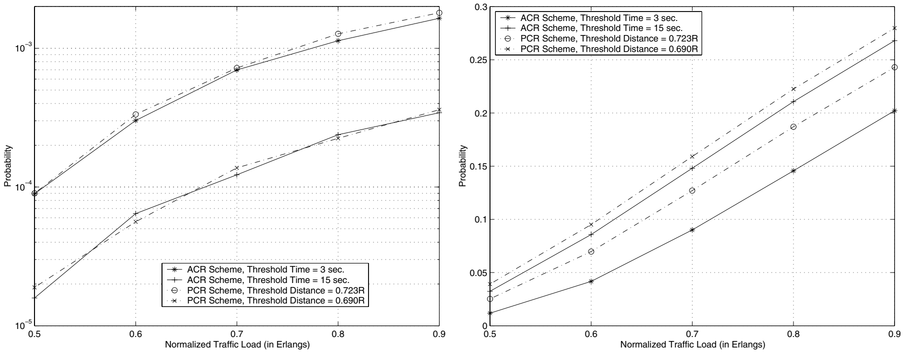
normalized traffic load, the ACR scheme allows a higher number of calls (approximately 4.5% more) than the PCR scheme, while maintains the same handoff call dropping probability P_d. Similarly, by choosing $T_{th} = 20$ seconds in the ACR scheme and $D_{th} = 0.69R_c$ in the PCR scheme, the two schemes have the same P_d. Correspondingly the value of P_b in the ACR scheme is lower by 1.5% as compared with that seen in the PCR scheme. Since the call incompletion probability P_{nc} is dominated by P_b, the ACR scheme can ensure more completed calls than the PCR scheme. Consequently, the ACR scheme achieves a higher channel utilization than the PCR scheme.

Figure 7 compares the performance of the two schemes under the low speed pattern V2. In Figure 7(a), by choosing $T_{th} = 3$ seconds and $D_{th} = 0.815R_c$, these two schemes have almost the same P_d for various normalized traffic loads. Correspondingly, in Figure 7(b), P_b in the ACR scheme is lower by 1% as compared with that in the PCR scheme. Similarly, by choosing $T_{th} = 20$ seconds and $D_{th} = 0.76R_c$, the two schemes have almost the same P_d, and P_b in the ACR scheme is lower by 1.5% as compared with that in the PCR scheme. One interesting observation is that P_b in the ACR scheme is much lower with a larger T_{th} under speed pattern V2 (In contrast, P_b is somewhat higher if T_{th} is larger as seen in Figure 6(a)). Under the low speed pattern, the average number of channel reservation requests is smaller than that under the high speed pattern, while the channel holding time of a call in a given cell is longer. This would imply that the rate at which the occupied channels are released will be smaller if we have a low speed pattern and the reservation request may therefore need a longer time to get an idle channel. For the same value of T_{th} and the same normalized traffic load, P_d under a low speed pattern is higher and P_b is lower than the corresponding values observed under a high speed pattern.

Figure 8 compares the performance of these two schemes under the high speed pattern. In Figure 8(a), by choosing $T_{th} = 3$ seconds and $D_{th} = 0.71R_c$ we ensures that the two schemes have almost the same P_d for various normalized traffic loads. In Figure 8(b), we see that P_b in the ACR scheme is lower by 4%

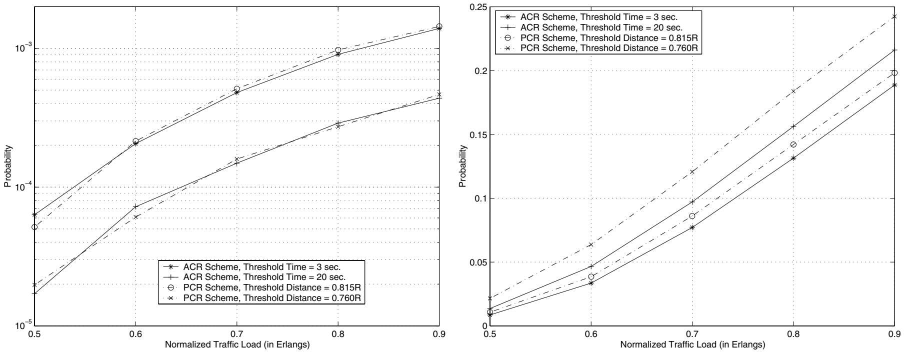
as compared with that seen in the PCR scheme. On the other hand, the ACR scheme has almost the same P_d and P_b as the PCR scheme when $T_{th} = 15$ seconds and $D_{th} = 0.6R_c$. This is due to the fact that under high speed patterns, if T_{th} is large, the channel reservation area will become very large; consequently the fraction of calls that make reservation requests in adjacent cells will be large. The performance of the ACR scheme will therefore deteriorate. However, even in this unrealistic scenario, the ACR scheme still performs as well as the PCR scheme.

Since the ACR scheme is distributed, it can be applied not only in homogeneous



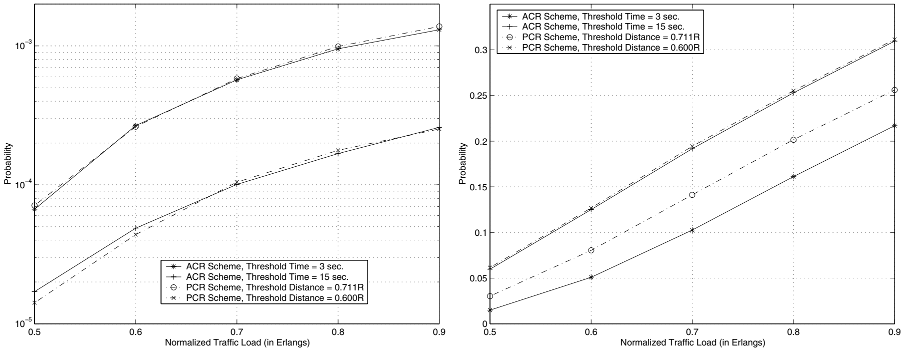
(a) P_d (b) P_b

Fig. 6. Performance comparison between the ACR scheme and the PCR scheme under V1



(a) P_d (b) P_b

Fig. 7. Performance comparison between the ACR scheme and the PCR scheme under V2



(a) P_d

(b) P_b

Fig. 8. Performance comparison between the ACR scheme and the PCR scheme under V3

systems in which every cell has the same size, shape and number of channels, but also in heterogeneous systems in which each cell might have a different coverage area, a different shape and different number of channels. The scheme may be expected to work well under non-uniform traffic loads as well.

5 Conclusion

In this paper, we propose an adaptive channel reservation (ACR) scheme for handoff prioritization which is based on GPS measurement. In this scheme, a base station sends a reservation request to a neighboring cell not only in accordance to the position and orientation of a mobile station, but also by taking into account the mobile station's relative moving speed with respect to its next target cell. The scheme introduces a new concept called the threshold time, and uses this in conjunction with other prior concepts such as reservation queueing, reservation cancellation and reservation pooling to minimize the effect of false reservations and to improve the channel utilization of the cellular systems. Extensive simulations were performed, and the simulation results show that, the ACR scheme can accommodate more new calls (has lower new call blocking probability P_b) than the PCR scheme while maintaining the same value of handoff call dropping probability P_d for any given traffic load.

References

1. S. Tekinay and B. Jabbari; "Handover policies and channel assignment strategies in mobile cellular networks," *IEEE Communications Magazine* Vol.29, No.11, 1991.
2. N.D. Tripathi, J.H. Reed and H.F. VanLandinoham; "Handoff in cellular systems" *IEEE Personal Communications* Vol.5, No.6, Dec. 1998.
3. D. Hong and S.S. Rappaport; "Traffic model and performance analysis for cellular mobile radio telephone systems with prioritized and non-prioritized handoff procedures" *IEEE Trans. on Veh. Tech.*, Aug. 1986

4. D. A. Levine, I. F. Akyildiz and M. Naghshineh; “A Resource Estimation and Call Admission Algorithm for Wireless Multimedia Networks Using the Shadow Cluster Concept.” *IEEE/ACM Transactions on Networking*, Vol. 5, No.1, February 1997
5. Y.C. Kim, D.E. Lee; “Dynamic Channel Reservation Based on Mobility in Wireless ATM Networks” *IEEE Communications Magazine*, Nov. 1999.
6. O. T. Yu and V. C. M. Leung; “Adaptive Resource Allocation for Prioritized Call Admission over an ATM-Based Wireless PCN” *IEEE Journal on Selected Areas in Communications*, Vol.15, No.7, September 1997
7. M.H. Chiu and M.A. Bassiouni; “Predictive Schemes for Handoff Prioritization in Cellular Networks Based on Mobile Positioning” *IEEE Journal on Selected Areas in Communications*, Vol.18, No.3, March 2000
8. W. Zhuang, K.C. Chua and S.M. Jiang; “Measurement-Based Dynamic bandwidth Reservation Sheme for Handoff in Mobile Multimedia Networks” *IEEE 1998 International Conference on Universal Personal Communications. Conference Proceedings*
9. B. Hofmann-Wallehnhof, H. Lichtenegger and J. Collins; “Global Positioning System: Theory and Practice”, Springer-Verlag, 1997.
10. B. Liang and Z.J. Hass; “Predictive Distance-Based Mobility Management for PCS Networks” *IEEE INFOCOM’99*
11. M. M. Zonoozi and P. Dassanayake; “User Mobility Modeling and Characterization of Mobility Patterns” *IEEE Journal on Selected Areas in Communications*, Vol. 15, No. 7, Sept. 1997.
12. H. Xie, S. Tabbane, and D.J. Goodman; “Dynamic Location Area Management and Performance Analysis” *Proceedings of 43rd IEEE Vehicular Technology Conference* May 1993.
13. W. Cui. and X. Shen; ”User Movement Tendency Prediction and Call Admission Control for Cellular Networks” *2000 IEEE International Conference on Communications. ICC 2000*
14. http://www.mesquite.com/ “CSIM18 Introduction”

Connection of Extruded Subnets: A Solution Based on RSIP

Cédric de Launois, Aurélien Bonnet, and Marc Lobelle

Université catholique de Louvain, Département INGI
Place Ste-Barbe, 2, 1348 Louvain-la-Neuve, Belgium
Fax: +32 10 45 03 45
delaunoi@info.ucl.ac.be, tel: +32 10 47 24 04
ab@info.ucl.ac.be, tel: +32 10 47 87 18
ml@info.ucl.ac.be, tel: +32 10 47 32 74

Abstract. Many remote computers need to be securely connected to their organization main network through a public IP network (e.g. Internet). Our purpose is to integrate as seamlessly as possible remote networks in the organization network, i.e. to put these in exactly the same situation as if they were located inside the organization. After summarizing the state of the art, the paper presents a solution based on RSIP, to dynamically allocate an IP address of the organization to a host of the remote network requesting an external access. Security is provided by IPSec. We compare this solution with a former proposal based on DHCP and show that the two solutions are very close but that RSIP brings us closer to an ideal situation but at an extra cost.

Introduction

Many organizations are faced to the problem of securely connecting remote computers to their network to accommodate nomadic users, teleworkers (including students in the case of educational institutions), remote branches and facilities etc. These remote systems are often connected to the main network of the organization through a public IP network, which can be the Internet or a provider network used to implement private virtual networks. In most instances, the remote computer obtains a single dynamic IP address in the provider range and security is added by encrypting the traffic in the application (SSL [1]), in an application tunnel (SSH [2]) or at IP level (IPSEC [3]).

In some situations these already classical solutions are inadequate.

One approach is to solve each individual problem when it appears. Another is to try to specify the ideal situation and to try to implement it. We choose this second approach. Our purpose is to integrate as seamlessly as possible remote machines in the organization network, i.e. to put these in exactly the same situation as if they were located inside the organization.

Since remote infrastructures often involve several computers (remote offices, student flats with several rooms, etc), we focus on remote subnetworks, rather

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 685–696, 2002.

than remote single computers. The latter case can always be considered as a particular case of the former.

We call subnets in this ideal situation "extruded subnets". They have the following properties (this is what we consider the ideal situation).

- The extruded subnet is connected to a gateway [1] that only needs a single dynamically allocated address in the provider range.
- Computers in the extruded subnet have addresses in the main network of the organization and are undistinguishable from computers located directly on the main network. In particular,
  - they can be used as clients as well as servers by any application;
  - no computer outside the company network and the extruded subnetwork can read, modify or detect traffic between a particular computer in the extruded subnetwork and the main network.
- Computers in the remote subnetwork have statically allocated permanent DNS names.
- Computers in the remote subnet use sparingly IP addresses, i.e. IP addresses are not allocated to computers that do not need it, e.g. that are either inexistent or stopped, etc.

We will first review the current techniques for connecting to a main network a subnetwork to which a temporary single IP address has been allocated. This review and the DHCP solution are based on [5] (where "extruded subnetworks" are called "remote bubbles"). Then the RSIP solution will be presented in detail. Finally the DHCP and RSIP based solutions will be compared and discussed.

1 Connection of a Subnetwork through a Single IP Address: State of the Art

This section presents the existing solutions to connect subnetworks through a single dynamically allocated IP address in ascending order of satisfaction of the requirements for extruded subnetworks.

1.1 Address Translation

NAT (Network Address Translation) is a solution where the gateway replaces the IP address in packets outgoing from the subnetwork by its own one, and the port number by one of its unused ones. The reverse substitution is performed on incoming packets. NAT allows client applications on computers of the subnetwork to invisibly access the Internet through the gateway. To other machines on the Internet, all this traffic will appear to be from or to the gateway (for more information see [6]). Not all "client" applications work with this scheme (e.g. FTP). It is therefore often complemented with specific application level proxies.

[1] We will use the generic name of "gateway" for the computer linking the subnetwork to the rest of the world

NAT is inadequate when the machines in the extruded subnet must be accessible from outside (e.g. for direct videoconference or for peer to peer applications).

1.2 Virtual Private Networks

VPNs have been introduced to let two networks communicate securely when the only connection between them is over a third network which they don't trust. VPNs use a gateway between each of the communicating networks and the untrusted network. Most current VPN packages use tunneling.

The gateway can encrypt packets entering the untrusted net and decrypt packets leaving it, in order to secure the tunnel.

Simple Tunneling Protocol. Most current operating systems can enable simple tunnels between two gateways (without any authentication or encryption : the tunnel is thus not secure). Gateways on each network encapsulate packets destined to the distant network in a packet destined to the remote gateway. Gateways identify each other using their static IP addresses. In our context, this gateway address is dynamically allocated by the provider. So it must be authenticated by other means than its IP address.

The IPSec Protocol and its Use in Virtual Private Networks. IPSec is a mechanism for adding security to IP. It can protect traffic between hosts, between network security gateways (routers, firewalls,. . .) and between hosts and security gateways. IPSec hosts and gateways are authenticated by cryptographic techniques independently of their IP addresses, which may be dynamically allocated. More informations can be found in [4], [3].

The VPN can be built by deploying IPSec gateways using IPSec in tunnel mode.

Current VPN solutions are inadequate in organizations that cannot afford to assign permanent IP addresses to machines in the remote subnets

1.3 Extruded Subnets

This first implementation is based on three different protocols : IPSec, DHCP (Dynamic Host Configuration Protocol), and NAT (Network Address Translation) or proxy ARP. More information is available in [5].

First step : Building Static IPSec VPNs. The first step is to set up an IPSec VPN between the gateway of the extruded subnet and a gateway in the main network like in the preceding solution.

This provides already the following features :

- The computer in the extruded subnet is logically neighbour of the main network.

- The external address of the gateway may be dynamically allocated.
- IPSec can provide security (authentication and confidentiality).

All the packets destined to the extruded subnets will be routed through the tunnels.

Second step : Adding Dynamic IP Address Allocation to the Computers in the Extruded Subnets. The Dynamic Host Configuration Protocol *(DHCP)* automates the process of configuring devices on IP networks. DHCP performs many of the functions a network administrator could carry out manually when connecting a new computer to a network (see [7]). With DHCP relay agents, remote machines can also be configured. A relay agent is used to forward DHCP messages between clients and server when the server and the client are not in the same network. The central DHCP server knows what set of IP addresses it must allocate to requests for each relay agent. The DHCP protocol with relay agents can be used to dynamically configure the computers of extruded subnetworks with the following advantages :

- the network configuration of the computer is easier (most of the parameters are transmitted by the protocol),
- addresses can be leased temporarily when needed, which, for instance, simplifies network administration of nomadic computers (laptops),
- subnetworks can be created without any administrative overhead for address allocation in the subnet,
- the DHCP protocol is available on many operating systems.

In this DHCP based implementation of extruded subnets, modified relay agents run on the gateways of the extruded subnets. The difference with standard relay agents is that addresses are assigned to the devices on the different subnets without regard to their localization. This solution is more economical in IP addresses, but routes must be explicitly configured for each individual device. When a device asks for a new DHCP configuration, the relay agent offers a dedicated IPSec tunnel opened between the gateway of the extruded subnet and one in the main network for this new IP address. The device has the illusion to be connected by a point-to-point link to the gateway in the main network.

Third Step. In the preceeding solution, all the packets sent by a device on an extruded subnet will be routed through the gateways even the packets destined to the subnet itself, as the different devices of our extruded subnet do not know they are on the same physical network.

Two different techniques may be used to give them that knownledge. In both, the machines on the extruded subnet are made to believe they are on a large network including all the remote subnets. In the first technique, instead of sending an address such as “a.b.c.d”, the DHCP server will send the private IP address “10.b.c.d” with the same three last bytes and a class A subnetwork mask instead of a point-to-point mask. This way, all devices in the extruded network can see each other. The gateway has to "NAT" (translate address) between 10.b.c.d and a.b.c.d for incoming and outgoing messages, with the aforementioned disadvantages of NAT.

In the other solution, the devices get a netmask covering the set of addresses allocatables to all the extruded subnets. Typically, when a machine on an extruded subnet wants to communicate with a machine on another extruded subnet, Proxy ARP on the gateway will answer so that all traffic to the remote machine will be sent to it. From there, it will be routed to the destination extruded subnet.

2 Using RSIP to Manage Address Allocation in Extruded Subnets

Another way to integrate computers on a remote subnet into the main network of an organization is using the new RSIP (*Realm Specific IP*) protocol [8], [9], [10]. RSIP has been designed as an alternative to NAT but with the additional requirement to preserve end-to-end packet integrity, a feature not provided by NAT. RSIP is based on the concept of granting a host (called RSIP host) from a network A a presence in another network, B, by allowing it to use resources (e.g. addresses and other routing parameters) from the network B. The gateway (called RSIP gateway) between networks A and B owns a pool of such resources, that it can allocate to RSIP hosts. For connecting a private network to a public one, a gateway on the boundary between these networks owns a pool of public IP addresses that it can allocate to hosts of its private network. See figure 1.

The problem of connecting an extruded subnet to a remote main network is similar since the previously described gateway may be split in two parts, connected via a tunnel. We will deal with the problem of the distance between these two networks in section 3. We may thus first focus on the simple problem of dynamically allocating IP addresses to hosts of a private network connected to a public one.

RSIP has been defined in two basic flavors : RSA-IP and RSAP-IP. When using RSAP-IP, the RSIP gateway maintains a pool of IP addresses as well as pools of port numbers per address. The gateway allocates each IP address with one or more port numbers. A host may only use the tuples address/port that have been assigned to it. When using RSA-IP, a RSIP gateway only maintains a pool of IP addresses to be leased by RSIP hosts. Upon request, the gateway allocates an address to the host, that may use it with any TCP or UDP port. This method is particularly interesting in our case and will be discussed below.

2.1 Using RSA-IP in Extruded Subnets

When a new computer is started in an extruded subnet based on RSIP :

- the computer boots with a private IP address;
- it registers with its RSIP gateway.

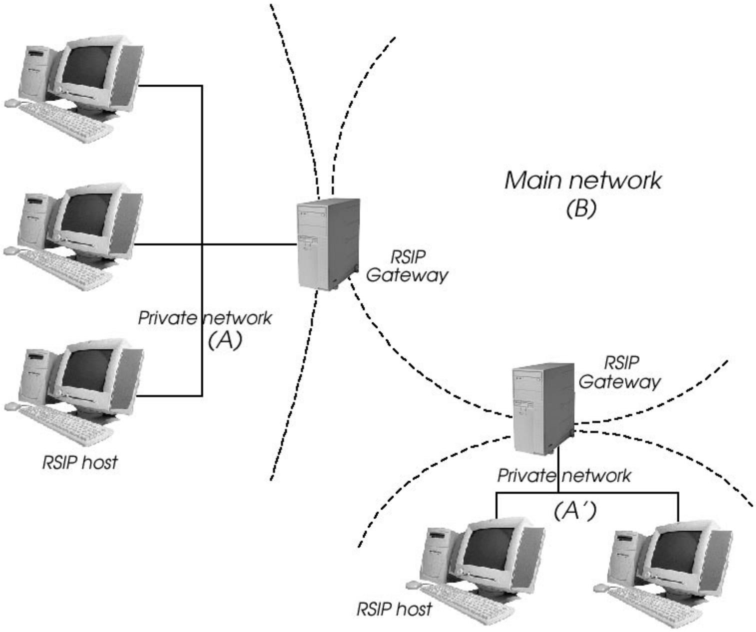


Fig. 1. Two private networks connected to the main network through RSIP gateways

- when it needs access to an external network, it requests an IP address from the gateway;
- the gateway delivers an address to the host, with an expiration time;
- the host uses this leased address for external accesses but still uses its private address to communicate with other hosts in the extruded subnet;
- when the lease time is about to expire, the host asks for a lease extension. If granted, the host may continue to use the address, otherwise it must release it.

2.2 The Routing Problem in RSIP-Based Extruded Subnets

Two main cases must be considered.

Communication between a Host X of the Extruded Subnet and a Host Y of an External Network (that may be another extruded subnet).

In this first case, the packets destined to hosts with public addresses must first be sent through the interface (often the interface of a tunnel to the gateway) corresponding to the public leased IP address. The gateway routes the packets it receives from X like regular packets. In the other direction, the packets originating from the public network can only be routed properly if the gateway is aware of the presence of a host with a public leased address inside the extruded subnet, and if a route or a tunnel to this host is available.

The RSIP gateway must establish this route each time it allocates an IP address to a host.

Communication between two hosts of two distinct extruded subnets is a particular case.

Communication between two Hosts X and Y in the Same Extruded Subnet. In this second case, when a host X wants to communicate with a host Y from the same extruded subnet, the routing will depend on whether host X contacts hosts Y using the leased public address of Y or not. If not, then host X will send its packets using its own private address (whether X possesses a leased address or not). No further routing is needed, Y will respond using its private address. If X uses the public address address of Y, the packets originating from X will reach the gateway first since it is the default gateway for IP packets destined to public hosts. Then, they will be routed to Y thanks to the special route established for packets coming from outside.

2.3 Using NAT/PAT in Coexistence with RSA-IP

The use of the RSA-IP protocol does not forbid to keep the NAT/PAT mechanism for hosts that only want to surf on the Internet or to use services for which a proxy exists.

The gateway must known which packets must be NAT'ed and which must not. The rule is to apply NAT only for packets with a private address in their header and destined to a public host (e.g. thanks to the iproute2 utility [11]).

3 Extension of the RSA-IP Protocol

The main drawback of the preceeding RSA-IP solution is, for our problem, its lack of scalability when there are several extruded subnets. RSA-IP, as described in [8], requires one pool of addresses per extruded subnet (the pool is kept on the gateway of each private network, see figure 1). Those addresses can thus only be allocated to hosts from that extruded subnet. This may lead to a waste of IP addresses if the range of addresses allocated to each remote subnetwork is statically allocated. It is much more efficient to maintain the pool in a unique, centralized, server. Moreover, the use of a central server makes maintenance and control a lot easier.

A way to obtain IP addresses from a central server is to use RSIP recursively : the RSIP gateways are themselves clients of a second level central RSIP server.

Another way, presented in this paper, is to extend the RSIP protocol (which is still an experimental) to make it support more possibilities.

A new agent is introduced in the system, the RSA-IP server, and is used to maintain, in a centralized way, the pool of addresses to be leased. The RSA-IP gateways are still located in the extruded subnets but don't own public addresses anymore : they are downgraded to proxies. See figure 2. A tunnel is established between the RSA-IP gateway and the RSA-IP server. Because of the use of this tunnel, the main network need not to be close to the extruded subnet. The tunnel may obviously be an IPSec tunnel.

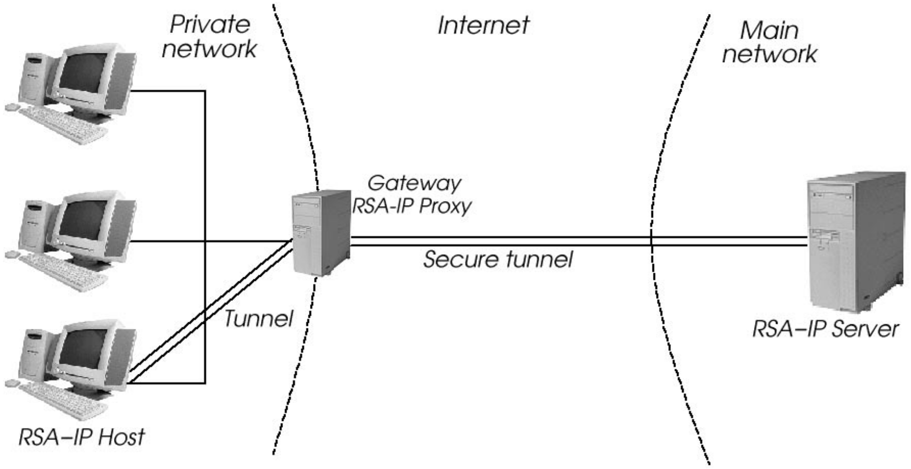


Fig. 2. A remote extruded subnet connected to the main network through a RSA-IP gateway and a RSA-IP server

A RSA-IP gateway just forwards requests from hosts of the extruded subnets to the server, which in turn replies just as if the requests were coming from a regular host. The specifications of this extension are beyond the scope of this paper. A prototype has been implemented [12].

For the purpose of dynamically allocating IP addresses to extruded subnets, the extension is equivalent to the recursive use of RSIP. However, the extension offers more possibilities, particularly for the dynamic binding of hosts to permanent domain names.

With this extension to RSIP, all the traffic to and from the extruded subnet travels through the tunnel. The Internet must thus route this traffic to the RSA-IP server and not directly to the gateway.

3.1 Binding Hosts in the Extruded Subnets to Permanent Domain Names

We want addresses in extruded subnets to be allocated dynamically but to be bound to permanent domain names. A partial solution is to let the server dynamically update the tables of the DNS server each time a resource is allocated to a host (e.g. using DNS Update protocol [13], [14]). However, this is only possible if the domain name of the host has been transmitted to the central RSIP server. This is not supported by the classical RSIP.

Moreover, at the time someone on the Internet tries to contact a server located in an extruded subnet, the latter may not yet have a dynamically leased public address. Those reasons led us to extend the classical RSIP solution.

The extension presented here and detailed in [12], proposes that a host sends its domain name when it registers with the central RSA-IP server. This way, when the host receives a leased IP address from the server, it becomes reachable by anyone using its domain name.

In addition, the extension also offers the possibility for a host to be warned (messages 2 and 3 in the figure 3) when it is contacted (message 1) by a public host, even if the contacted host has not yet requested a public IP address. In this case, the RSA-IP host may then request a public IP address from its gateway (messages 4 and 5) in order to be reachable by its correspondent. The solution is based on a two-way communication between the central RSA-IP server and the dynamical DNS server (messages 2 and 6).

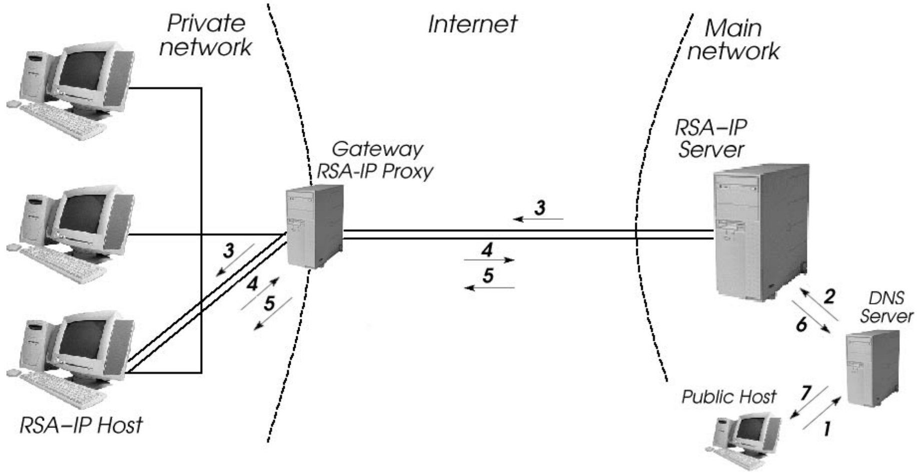


Fig. 3. Messages exchanged when a public host contacts a RSA-IP host which is not yet leasing an IP address

Thanks to the extension described above, DNS requests related to RSA-IP hosts can be handled through cooperation between RSIP and DNS servers.

This problem will not be further discussed in this paper.

4 Comparison between DHCP-Based and RSIP-Based Implementations of Extruded Subnets

We will compare in this section the use of the RSIP protocol instead of the DHCP protocol for the dynamic allocation of the public IP addresses to the hosts of the remote extruded subnets.

The services offered by RSA-IP and by DHCP are very close. However, RSA-IP provides somewhat different functionalities. For example, a RSIP gateway may be a policy enforcement point. In other words, it may have the ability to explicitly control which local addresses and ports are used to communicate with remote addresses and ports.

Both RSIP and DHCP have functionalities that can be used to spare IP addresses : RSA-IP as described in [8] and DHCP allow to specify expiration times for each allocated IP address. When this time expires, the RSA-IP or DHCP client may ask to extend its lease time. The RSA-IP gateway or the DHCP server may accept this extension or not. Thanks to these functionalities, we can dynamically allocate public IP addresses to hosts for the time they really need. This mechanism allows the saving of IP addresses.

DHCP and RSIP solutions are similar from the point of view of centralized address allocation. Both have been designed for on demand temporary allocation of IP addresses to hosts, but not to hosts in extruded subnets. However, both protocols can be used for this purpose.

A significant difference between DHCP and RSIP is how a host communicates with the server that allocates the addresses (DHCP server or RSIP gateway) when this server is on another physical network. A DHCP host communicates with the DHCP server through a DHCP relay agent located on its network because it has no IP address when the DHCP transaction is started. A RSIP host has already a local IP address when it starts the transaction and obtains a second (public) one through a RSIP transaction with the RSIP gateway. For this address to be useable, special routes must be set up (theoretically in the whole world) towards this address. Instead, a tunnel is usually set up between the host and the gateway and all external routes to RSIP hosts beyond a RSIP gateway point to this gateway. It must be noted that, before requesting an IP address, a RSIP host has to register with the RSIP gateway.

Thanks to relay agents, DHCP is very scalable regarding to the number of subnets. Besides, relay agents require no management. However, if only one relay agent is used in each subnet (i.e. the gateway), only simple networks (e.g. one ethernet) can be supported in the subnet. On the other hand, “standard RSIP” has been designed to allocate external addresses in large networks with any number of routers etc. So the scalability of RSIP is excellent regarding the size of the subnets, but RSIP has not been designed for handling several subnets. This means that each gateway must be managed by hand. This problem is solved either by using RSIP recursively or by the extension proposed to RSIP : the central RSIP server. With this, RSIP gets the same scalability as DHCP regarding the number of subnetworks.

DHCP has one significant advantage over RSIP : it is a well known and widely used protocol available on many operating systems. No special software must be added on the hosts of the extruded subnets. This is not the case with RSIP : software must be added on the RSIP hosts with current operating systems releases.

RSIP is designed to allocate a supplementary (external) IP address to a machine that has already an internal one. So local traffic can use the local address and external traffic can use the external address. Note that if RSIP is used to obtain the external address, DHCP can be used to obtain the local address.

DHCP is designed to allocate a first IP address to a machine. This single address serves two purposes : local and external. This brings a problem when addresses are allocated randomly in a set of subnets : when one cannot distinguish a machine on one's own subnet and on a remote subnet which makes routing between subnets impossible, proxy ARP creates the illusion that all subnets of a main network constitute a single network and remove the need of routing between them. The effect is the same as with the double address of RSIP.

Both the DHCP and RSIP based solution use IPSec tunnels to satisfy the security requirements of extruded subnets.

From a performance point of view, both solution are equivalent. They only differ in the address allocation to the hosts of the remote subnet, which is a relatively unfrequent operation without performance impact. During regular operation, the only overhead is that induced by IPSec. This is similar to what happens with any VPN. This overhead is negligible for ADSL and cable modem connections.

5 Conclusion

The problem exposed in this paper is to build extruded subnets, which are remote subnetworks virtually imported in another network, with IP addresses belonging to this network allocated on demand only to hosts that need it when they need it.

The solution proposed is based on the new RSIP protocol, modified to extend its functionality in order to manage IP adresses exported in the extruded subnets dynamically and centrally from the main network. Computers in the extruded subnets and others in the main appear to be in the same network.

The RSIP based solution has several advantages over the previous one based on DHCP. It has a better scalability regarding the size of the subnets and offers extended possibilities concerning the binding of RSA-IP host to permanent domain names : a RSA-IP host may be assigned an address at the time it is contacted by an external client. The extra cost of the RSIP solution is the necessity to add a software agent on each host in the extruded subnet. This agent must be designed for each operating system.

When these advantages are not useful, the DHCP solution is to be preferred since it does not require extra software on the client machines.

Acknowledgements. This work has been partially supported by grant FIRST 981/3749 by MRW-DGTRE ("Travaux réalisés avec l'aide du Ministère de la Région Wallonne, en application du programme de formation et d'impulsion à la recherche scientifique et technologique") and by a grant from FRIA (Fond pour la Formation à la Recherche dans l'Industrie et dans l'Agriculture, rue d'Egmont 5 - 1000 Bruxelles).

The definition and implementation of the extension to the RSA-IP is a common work of C. de Launois, G. Fauveaux and J. Honlet.

The authors express their thanks to the anonymous reviewers for their useful comments.

References

1. Freier, A.O.; Karlton, P.: Kocher P.C.; "The SSL Protocol Version 3.0", Internet Draft <draft-freier-ssl-version3-02.txt>, 1996.
2. Ylonen, T.; Kivinen, T.; Saarinen, M.; Rinne, T.: Lehtinen, S.: "SSH Protocol Architecture", Internet Draft <draft-ietf-secsh-architecture-09.txt>, work in progress, July 2001.
3. Kent, S.; Atkinson, R.: "Security Architecture for the Internet Protocol. Network Working Group", RFC 2401, 1998.
4. Doraswamy, N.; Harkins, D.: "IPSec : The New Security Standard for the Internet, Intranets and Virtual Private Network", Prentice Hall PTR, 1999.
5. Bonnet, A.; Lobelle, M.: "Extending a Campus Network with remote Bubbles using IPSec", I-NetSec'01, Leuven, 2001.
6. Ranch, D.: "Linux IP Masquerading HOWTO. Technical Report", 2000.
7. Droms, R.; Lemon, T.: "The DHCP Handbook", Macmillan Technical Publishing, 1999.
8. Borella, M.; Grabelsky, D.; Lo, J.; Taniguchi, K.: "Realm Specific IP : Protocol Specification", RFC 3103, October 2001.
9. Borella, M.; Lo, J.; Grabelsky, D.; Montenegro G.: "Realm Specific IP : Framework", RFC 3102, October 2001.
10. Montenegro G.; Borella, M.: "RSIP Support for End-to-end IPSEC", RFC 3104, October 2001.
11. Kuznetsov A. N.: "IP Command Reference", Institute for Nuclear Research, Moscow, April 1999.
12. de Launois C.; Fauveaux G.; Honlet J.: "Routeur d'accès à adresse dynamique", Université catholique de Louvain, Louvain-la-Neuve, 2001. http://openresources.info.ucl.ac.be/rsip/
13. Eastlake, G.: "Secure Domain Name System Dynamic Update", RFC 2137, April 1997.
14. Wellington, B.: "Secure Domain Name System (DNS) Dynamic Update", RFC 3007, November 2000.

Adjusted Probabilistic Packet Marking for IP Traceback

Tao Peng[1], Christopher Leckie[1], and Kotagiri Ramamohanarao[2]

[1] ARC Special Research Center for Ultra-Broadband Information Networks
Department of Electrical and Electronic Engineering
The University of Melbourne
Victoria 3010, Australia
{t.peng,c.leckie}@ee.mu.oz.au
http://www.ee.mu.oz.au/cubin

[2] Department of Computer Science and Software Engineering
The University of Melbourne
Victoria 3010, Australia
rao@cs.mu.oz.au

Abstract. Distributed denial-of-service attack is one of the greatest threats to the Internet today. One of the biggest difficulties in defending against this attack is that attackers always use incorrect, or "spoofed" IP source addresses to disguise their true origin. In this paper, we present a packet marking algorithm which allows the victim to traceback the approximate origin of spoofed IP packets. The difference between this proposal and previous proposals lies in two points. First, we develop three techniques to adjust the packet marking probability, which significantly reduces the number of packets needed by the victim to reconstruct the attack path. Second, we give a detailed analysis of the vulnerabilities of probabilistic packet marking, and describe a version of our adjusted probabilistic packet marking scheme whose performance is not affected by spoofed marking fields.

1 Introduction

Distributed denial-of-service (DDoS) attacks have become a major threat to the Internet [10]. At the same time, DDoS is extremely difficult to defend [6]. The reason lies in the fact that the attackers use incorrect ("spoofed") IP addresses in the attacking packets and therefore disguise the real origin of the attacks. This has made it very difficult or impossible to traceback the source of attacking IP packets.

A number of recent studies have approached the problem of IP packet traceback by Probabilistic Packet Marking (PPM) [15] [17]. It is assumed that the attacking packets are much more frequent than the normal packets. The main idea is to let every router mark packets probabilistically and let the victim reconstruct the attack path from the marked packet. All of the probabilistic marking algorithms try to overload the marking information into the 16 bit identification field in the IP packet header, which is seldom used [5] [18]. A major issue with

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 697–708, 2002.

existing probabilistic marking schemes is that they use a fixed marking probability, which means that there is a greatly reduced probability of getting packets from routers which are far away from the victim. Consequently the number of packets needed to reconstruct the attack path depends on the number of packets which are marked by the furthest router in the attack path. If we can increase the marking probability for the routers which are far away from the victim, then we need less packets to reconstruct the attack path.

A potential problem with packet marking is that the attacker can forge the marking field. The authenticity of the marking field is the biggest challenge for Probabilistic Packet Marking, which is discussed in [13]. Although Song and Perrig [17] have proposed a scheme for router authentication, it is still hard to implement and there are still some chances for the attacker to spoof the marking field. However, if we can let the routers mark all the packets when they first enter the network, then there is no way for the attacker to use the spoofed marking field to decoy the victim.

In this paper, we make two contributions to the technique of Probabilistic Packet Marking. First, we have developed three techniques for adjusting the probability used by routers to mark packets, in order to reduce the number of packets needed by the victim to reconstruct the attack path. Second, we give a detailed analysis of the vulnerabilities of PPM, and describe a version of our adjusted probabilistic packet marking scheme whose performance is not affected by spoofed marking fields. We demonstrate the benefits of our approach with an analytical model as well as providing an experimental evaluation using simulated packet traces.

The paper is organized as follows. We present a brief background to this problem and highlight the main challenges of IP marking in Section 2. Section 3 introduces our Adjusted Probabilistic Marking Algorithm and shows a theoretical analysis. Simulation results of all these three techniques are provided in Section 4. From the analysis and simulation results, we can see that our Adjusted Probabilistic Marking Algorithm is more efficient and secure than the previous marking schemes. We discuss some practical issues in Section 5 and the related work is given in Section 6. Finally we conclude in Section 7.

2 Background on Probabilistic Packet Marking (PPM)

Once an attack has been detected, an ideal response would be to block the attack traffic at its source. Unfortunately, there is no easy way to track IP traffic to its source. This is due to two features of the IP protocol. The first feature is the ease with which IP source addresses can be forged. The second feature is the stateless nature of IP routing, where routers normally know only the next hop for forwarding a packet, rather than the complete end-to-end route taken by each packet. This design decision has given the Internet enormous efficiency and scalability, albeit at the cost of traceability. In order to address this limitation, Probabilistic Packet Marking (PPM) has been proposed to support IP traceability.

2.1 Definitions

The main idea of PPM is to let routers mark the packets with path information probabilistically and let the victim reconstruct the attack path using the marked packets.

Denial-of-service attacks are only effective so long as they occupy the resources of the victim. As a result, most denial-of-service attacks are comprised of thousands or millions of packets. PPM is based on the assumption that when we mark each packet with only a small probability then the victim will receive sufficient packets to reconstruct the attack path.

The network can be viewed as a directed graph $G = (V, E)$ where V is the set of nodes and E is the set of edges. V can be further partitioned into end systems (leaf nodes) and routers (internal nodes). The edges denote physical links between elements in V. Let $S \subset V$ denote the set of attackers and let $t \in V/S$ denote the victim. We will first consider the case when $|S| = 1$ (single-source attack) and treat the distributed DoS attack case separately. We assume that routes are fixed, and that the attack path $\mathrm{A} = (s, v_1, v_2, ..., v_d, t)$ is comprised of d routers (or hops) and has path length d [13].

Let N denote the number of packets sent from s to t. A packet x is assumed to have a marking field where the identity of a link $(v, v') \in E$ traversed can be inscribed. A packet travels on the attack path sequentially. At a hop $v_i \in \{v_1, ..., v_d\}$, packet x is marked with the edge value $(v_{i-1}, v_i), i = 1, ..., d,$with probability p. As we seen in Fig.1, packet 1 is marked with edge value (v_1, v_2) and distance 2; packet 2 is marked with edge value (v_2, v_3) and distance 1. When t receives two packets it can reconstruct the attack path (v_1, v_2, v_3).

Each router marks a packet with probability p. When the router decides to mark a packet, it writes its own IP address into the edge field and zero into the distance field. Otherwise, if the distance field is already zero, which means this packet has been marked by the previous router, it processes the packet as follows: (1) It combines its IP address and the existing value in the edge field and writes the combined value into the edge field. (2) It increases the distance value by 1. Thus, the edge value contains both information from the previous router and the current router. Finally if the router does not mark the packet, then it always increments the distance field. This distance field indicates the number of hops between the victim and the router that has marked the packet. The distance field should be updated using saturating addition, meaning the distance field is not allowed to wrap. When using this scheme, any packet written by the attacker will have a distance field greater than or equal to the real attack path. In contrast, a packet which is marked by the router should have a distance field which is less than the length of the path traversed from that router.

Savage et al. propose a method called Fragment Marking Scheme (FMS) [15] to compress the IP addresses and recontruct the attack path. It is later improved by Song and Perrig[17]. Unless otherwise stated, when we talk about PPM in the rest of this paper, we are referring to Song and Perrig's version of PPM.

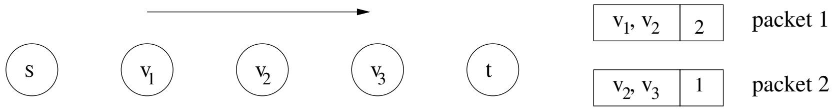


Fig. 1. Probabilistic Packet Marking

2.2 Limitation of Previous PPM Schemes

Our aim is to minimize the time required to reconstruct the attack path. This depends on the time it takes to receive packets that have been marked by each router on the attack path. This in turn depends on the choice of the marking probability p. In this section, we model the performance of PPM in terms of p, and highlight the limitation of using a fixed marking probability.

Definition 1. Let α_i denote the probability that packet arriving at the victim is lastly marked at node v_i but nowhere after v_i. For a uniform marking probability, $\alpha_i = \Pr\{x_d = (v_{i-1}, v_i)\} = p(1-p)^{d-i}$ $(i = 1, 2, \ldots, d)$.

Definition 2. Let α_0 denote the probability that a packet sent from the attacker reaches the victim without being marked at any of the routers. For a uniform marking probability, $\alpha_0 = (1-p)^d$. In order to reconstruct the attack path as quickly as possible, the victim needs to receive a sample of packets marked by each router in the path. An unmarked packet provides no information to the victim. In fact, there is a risk that unmarked packets may contain misleading information that has been spoofed by the attacker. Consequently, we want as many packets to be marked as possible. This implies that p should be large, so that α_0 is as small as possible. However, there is a penalty for making p too large. As p increases, there is a greater likelihood that packets marked by routers close to the source will be overwritten by routers close to the victim. Note that $\alpha_d \geq \ldots \alpha_2 \geq \alpha_1$, so α_1 is the smallest value. This is worst for packets marked by the first router after the source. So we need to choose p such that α_0 is minimized and α_1 is maximized.

According to the *coupon collecting problem* [8], for each attack path with d routers (excluding the victim), and with marking probability p, the expected number of packets needed to reconstruct the attack path is $N(d) = \frac{\ln(d)+O(1)}{p(1-p)^{d-1}}$ [15].

We can show that $N(d)$ is minimised when $p = \frac{1}{d}$. Consequently, the number of packets needed to get one sample from each router is $N_f(d) \simeq \frac{\ln(d)+O(1)}{\frac{1}{d}(1-\frac{1}{d})^{d-1}}$. This is the best result we can achieve for marking algorithms with a fixed probability. Our proposal is to reduce the total number of packets required $N(d)$ by using a higher marking probability for routers close to the source. Ideally, we want to receive an equal number of packets marked by each router on the attack path, i.e. $\alpha_i = 1/d$. In this case, the number of packets needed for reconstruction is $N_a(d) = d\ln(d)$. The savings of this approach are $\frac{N_f(d)}{N_a(d)} = (1-\frac{1}{d})^{1-d}$, which is greater than 2 for $d \geq 2$. Our aim has been to develop a technique for adjusting the marking probability so that we can achieve the performance of $N_a(d)$.

3 Adjusted Probabilistic Packet Marking Schemes

According to the analysis in section 2, we propose that a router should adjust its packet marking probability based on its position in the attack path. However, the position of the router in the attack path is not known, since the position of the attacker is unknown. We need to estimate this distance based on the available information. In this section, we propose 3 different schemes for adjusting the marking probability based on the different distance measures d_1, d_2 and d_3. The definition of these distances is shown in Fig. 2

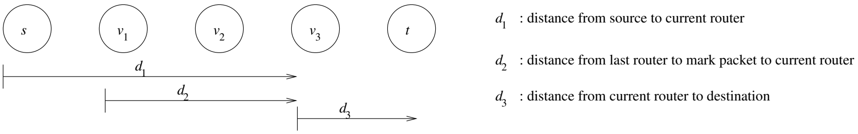

s
v_1
v_2
v_3
t
d_1
d_2
d_3
d_1 : distance from source to current router
d_2 : distance from last router to mark packet to current router
d_3 : distance from current router to destination

Fig. 2. Definitions of different distance measures

3.1 Number of Hops Traversed by Packet d_1

Let every router mark the packet with probability $p_1(d_1) = 1/d_1$. The ideal case for packet marking is to receive packets marked by each router with equal probability $\alpha_i = 1/d$, if the path length is d. Let $p_1(d_1)$ represent the marking probability of the router at distance d_1 from the source, where $d_1 = 1, 2, ..., d$. Then we obtain the following equations:

$$\alpha_d = p_1(d) = 1/d \tag{1}$$

$$\alpha_{d-1} = p_1(d-1)[1 - p_1(d)] = 1/d \tag{2}$$

$$\alpha_{d-2} = p_1(d-2)[1 - p_1(d-1)][1 - p_1(d)] = 1/d \tag{3}$$

From equation 1 we can get $p_1(d) = 1/d$; from equation 2 we can get $p_1(d-1) = 1/(d-1)$; from equation 3 we can get $p_1(d-2) = 1/(d-2)$. Accordingly, we can summarize the marking probability formula as $p_1(d_1) = 1/d_1$. Then for the router at distance d_1, $\alpha_{d_1} = \frac{1}{d_1} \times (1 - \frac{1}{d_1+1}) \times (1 - \frac{1}{d_1+2}) \times ... \times (1 - \frac{1}{d})$. This equation can be simplified as $\alpha_{d_1} = \frac{1}{d_1} \times \frac{d_1}{d_1+1} \times \frac{d_1+1}{d_1+2} \times ... \times \frac{d-1}{d} = \frac{1}{d}$. This means if each router marks the packet with the probability $p_1(d_1) = 1/d_1$, we can receive the packets marked by each router with equal probability $1/d$, given the path length is d.

In order to implement this marking scheme, we need to know the distance measure d_1. We propose to add an extra field in the IP option field. This field can be used to record the number of hops (d_1) traversed by the packet. The default value for this field is 0, and the router increases this value by 1 every time it forwards the packet. Every time the router gets the packet, it extracts the information d_1 from the option field and marks the packet with probability $1/d_1$. In order to prevent the attacker from spoofing this field, we can use the encryption schemes which are discussed in [17].

3.2 Number of Hops Traversed Since the Packet Was Last Marked (d_2)

In the original Probabilistic Packet Marking (PPM) scheme [15], there are three parts in the marking field. One part is called the distance field (d_2), which is used to hold the distance information from last router to mark the packet to the current router. We denote $d_2 = 0$ for routers next to each other. Let each router mark the packet according to the formula: $\frac{1}{2(d_2+1)}$. Since the larger the d_2 value, the higher the likelihood that it will be overwritten. Thus, we believe we should use a low marking probability for the packets with high d_2 value. Let us now illustrate the derivation of this formula by considering an example when the attack path length is 3.

The router marks the packet which has a distance value d_2 in the marking field with a probability $p_2(d_2)$. We assume the routers mark each packet when it first enters the network. So when the packet passes the first router, the d_2 value will be set to 0. By analyzing all the possibilities of the d_2 value when the packets traverse the attack path, we can derive expression for $\alpha_i, i = 1, 2, \ldots, d$. Using these equations, we can find optimal marking probabilities for $\alpha_1, \alpha_2, \alpha_3$. However, the equations become more complicated as the path length increases, we consequently propose that the general marking probability should be $p_2(d_2) = \frac{1}{2(d_2+1)}$, which has been shown through experiments to have the best performance.

Since there are 5 bits in the marking field to hold the information in the existing probabilistic marking scheme [15] [17], we only need to extract this information from the marking field and mark the packet according to the formula $p_2(d_2) = \frac{1}{2(d_2+1)}$.

3.3 Number of Hops from Current Router to Destination (d_3)

If we can get the distance of the current router to the destination (d_3), we can mark each packet with a probability $p_3(d_3) = 1/(c+1-d_3)$ where c is a constant, and then we can receive packets marked by each router with a probability of $1/c$.

According to the marking scheme, we can have $\alpha_{d_3} = \frac{1}{c+1-d_3}(1 - \frac{1}{c-d_3+2})...(1 - \frac{1}{c-1})(1 - \frac{1}{c}) = \frac{1}{c+1-d_3} \times \frac{c-d_3+1}{c-d_3+2} \cdots \frac{c-2}{c-1} \times \frac{c-1}{c} = \frac{1}{c}$. In order to make this scheme work, we have to make sure $c + 1 - d_3 > 0$. Since most path lengths in the Internet are bounded by 30 [4] [1] [19], we can take $c = 30$ for safety. So if we mark with probability $p_3(d_3) = 1/(31 - d_3)$, we can make sure we can receive the packets marked by each router with probability 1/30.

We rely on the routing protocol to provide us with the distance measure d_3. Current Internet routing protocols are destination-based and every time the router forwards the packet, it will look at the routing table to find the next hop to the destination. Internet protocols provide us with a measure of the number of hops to each destination, which can be stored in the routing table as a measure of distance d_3. When the router starts to route the packet, it can extract the distance information d_3 from the routing table and then mark the packet according to the formula $p_3(d_3) = 1/(31 - d_3)$.

3.4 Summary

We can summarize each marking scheme in term of its performance and practicality.

Marking scheme 1: $p_1(d_1) = 1/d_1$ can achieve the ideal marking performance. With this marking scheme, we can receive the packets marked by each router with equal probability for path length. Furthermore, every packet is marked under this scheme, and the attacker has no chance to spoof the marking field. However, this scheme requires a special hop count field and there is a risk that this field can be spoofed by the attacker. In order to make this scheme work, we need a strong authentication scheme which can stop the attacker from spoofing, e.g. [17].

Marking scheme 2: $p_2(d_2) = \frac{1}{2(d_2+1)}$ uses the distance field that is part of the packet marking scheme. This scheme can achieve a performance which is close to the optimal performance. In order to make this scheme work, we need to make sure the distance value in the marking field is trustable. One possibility is to let the routers mark all the packets when they first enter the network, then the attackers have no way to spoof the distance value. However, this is only practical if we control the ingress routers to our network, and thus is effectively the same as a technique called ingress filtering [9].

Marking scheme 3: uses information from the routing protocol and can achieve better results than using the uniform marking probability. Since the information is from the routing protocol, it can not be manipulated by an attacker. So scheme 3 is the safest and most practical scheme.

4 Evaluation

Our aim is to compare the performance of each scheme to PPM. Our comparison is based on the number of packets needed to reconstruct an attack path for a range of simulated attacks.

4.1 Methodology

We simulate attacks from different distances using the methodology in [17]. The network topology is based on a real traceroute dataset obtained from Lucent Bell Labs [11]. In our simulation, we vary the attack path from 1 to 30 hops and conduct 1000 random trials at each path length value. We measured the number of packets required to reconstruct the attack path using our schemes, and compared this to the number of packets required by PPM [17], where our implementation of PPM used a threshold of M=5 as defined in [17]. We varied the uniform marking probability of PPM using the values $p = 0.01, 0.04$, and 0.1. Note that $p = 0.04$ is recommended as the optimum choice for PPM [15].

4.2 Results

The performance of schemes 1 to 3 are shown in Fig. 3.

Schemes 1 and *2* perform the best, outperforming PPM for all values of p tested. However, these results assume that the distance field has not been tampered with. *Scheme 3* is the most practical, since its distance measure cannot be tampered with by the attacker.

Scheme 3 outperformed PPM with $p = 0.01$ and 0.04. Although PPM with $p = 0.1$ outperforms *scheme 3* for small hop counts, *scheme 3* performs far better when the attack path is large.

Scheme 3 outperforms *scheme 2* when path length is 20 or higher as shown in Fig. 3. This is because as the path length increases, *scheme 3* approaches optimum performance while *scheme 2* cannot achieve the optimum performance as we discussed in Section 3.2. Furthermore, *scheme 1* and *scheme 3* converge when the path length equals 30 because c equals the path length, which makes $p_3(d_3)$ equivalent to $p_1(d_1)$.

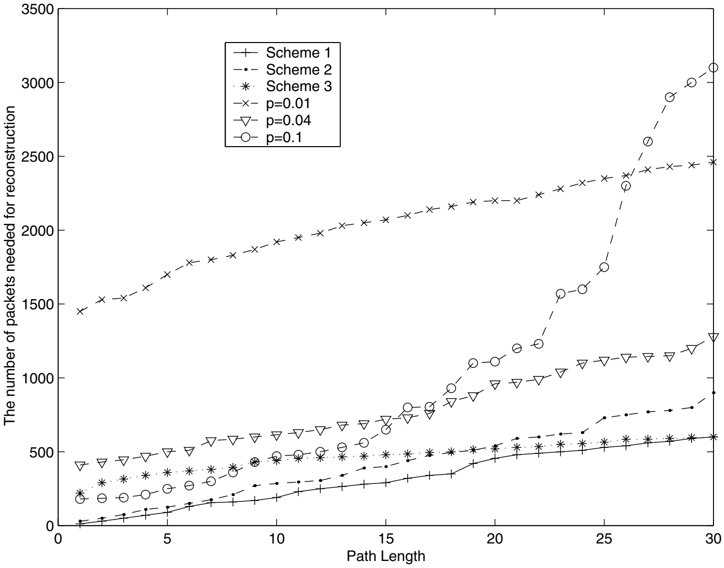


Fig. 3. *Scheme 1,2,3* compared with uniform marking probability

5 Discussion

5.1 Distributed Denial-of-Service Attacks

During a distributed denial-of-service attack, there are many attacking sources. We have found that the number of packets needed for reconstruction increases linearly with the number of attackers. So it will become very hard to verify all the attacking sources during a DDoS attack. Thus, our method to reduce the number of packets needed for reconstruction becomes extremely important to improve the reconstruction efficiency.

5.2 Spoofing the Marking Field

By spoofing the marking field, it is possible for attackers to make the attack appear as though it has come from a more distant source, e.g. a false source s_f as shown in Fig. 4. However, the attacker cannot change the marking of routers between it and the victim, e.g., v_1 to v_3. Consequently, we can always reconstruct the path to the attacker, although we may also reconstruct a false sub-path at the start of the true path, e.g., v_{f_1} to v_{f_3}.

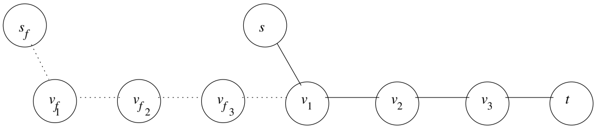


Fig. 4. Effect of Spoofing the Marking Field (Fake sub-path: v_{f_1} to v_{f_3},true path: v_1 to v_3)

If we are unable to authenticate the marking field, then this false sub-path can affect the performance of our first two schemes. This is because distance measures d_1 and d_2 will be inflated by the false sub-path, thus decreasing the packet marking probability of routers in the true attack path.

However, our third scheme is unaffected by the actions of the attacker. This is because d_3 is derived from information in the routing table of each router, and the destination field. The attacker cannot fake the destination field without defeating the purpose of the attack, and the attacker cannot manipulate the contents of the routing tables in the routers. Thus, the performance of our third scheme is secure against manipulation by the attacker.

6 Related Work

Burch and Cheswick [3] propose a link-testing traceback technique. It infers the attack path by flooding the links with large bursts of traffic and observing how this perturbs the attack traffic. This scheme requires considerable knowledge of network topology and the ability to generate huge traffic in any network links. Mahajan et al. [12] provide a scheme in which routers learn a congestion signature to tell good traffic from bad traffic. The router then filters the bad traffic according to this signature. Furthermore, a pushback scheme is given to let the router ask its adjacent routers to filter the bad traffic at an earlier stage. This scheme is effective for some types of DDoS attacks but it needs a narrow and accurate congestion signature to make sure the bad traffic is filtered while the good traffic is not affected.

Bellovin [2] proposed an ICMP "traceback" scheme to let router generate ICMP packets to the destination containing the address of the router with a low probability. For a significant traffic flow, the destination can gradually reconstruct the route that was taken by the packets in the flow. ICMP packets are often treated with a low priority by routers to reduce the additional traffic, which undermines the effectiveness of the scheme. This scheme is later extended by Wu et al. [20]. An alternative approach is to mark the packets themselves. Savage et al. [15] describe a scheme for routers to probabilistically mark packets. They propose using the identification field of the IP header, which is normally used to control fragmentation. They point out that IP fragmentation is seldom used in practice. While their approach overcomes many of the limitations of the ICMP traceback proposal, there are some security problems when the attackers fake the marking field. Song et al. [17] propose an enhanced scheme of probabilistic packet marking and also set up a scheme for router authentication. However, the authentication scheme is complex to implement. Dean et al. [7] propose an alternative marking scheme using noisy polynomial reconstruction. This scheme is backwards compatible, and incrementally deployable compared with the former proposals. Unfortunately their scheme is very vulnerable to fake markings put in the packets by the attackers. Furthermore, the number of packets needed to reconstruct the attack path is quadratic to the number of attackers. Snoeren et al. [16] propose a scheme to let routers store a record of every packet passing through the router, so that the router can then trace back the origin of the packet by using the history in the router. Although they describe a smart scheme to compress the storage, it is still a huge overhead for the router to implement this scheme, especially with the increasing network speed. Park and Lee [14] propose to put distributed filters in the routers and filter the packets according to the network topology. This scheme can stop the spoofed traffic at an early stage. However, in order to place the filters effectively, it needs to know the topology of the Internet and routing policy between Autonomous Systems, which is hard to achieve in the expanding Internet.

In summary, every marking scheme uses a fixed marking probability which will result in a small number of packets marked by the more distant routers when all the packets arrive at the victim. In contrast, we have developed several schemes that solve this problem by adjusting the marking probability in each router, which significantly reduces the number of packets required to reconstruct the attack path. Furthermore, no one has set up a scheme to completely solve the security problem that the attacker can fake the marking field. However, our third marking scheme does not use the contents of the marking field to adjust the marking probability, and thus cannot be manipulated by the attacker while at the same time requiring fewer packets to trace the packet.

7 Conclusion

In this paper, we make the following two contributions to Probabilistic Packet Marking (PPM). First, we developed three techniques to adjust the marking probability used by each router so that the victim receives packets marked by each router with equal probability. *Scheme 1* is to let the IP packet carry a message to inform the router how far the packet has traveled. *Scheme 2* is to use the distance value of the marking field in the IP packet. *Scheme 3* is to get the distance between the router and destination from the routing table. Both *scheme 1* and *scheme 2* need authentication to prevent the attacker from spoofing the required information. *Scheme 3* is the most practical one and can improve the reconstruction efficiency compared with the optimal uniform marking probability ($p = 0.04$). By implementing this scheme, we can substantially reduce the number of packets needed to reconstruct the attack path in comparison to PPM. Our second contribution is that we give a detailed analysis of the vulnerability of PPM, and describe a version of our adjusted probabilistic packet marking scheme whose performance is not affected by the vulnerability caused by spoofed marking fields.

Acknowledgment. We would like to thank the AT&T Internet Mapping Project for making available their traceroute data and the anonymous reviewers for their helpful comments.

References

1. Skitter Analysis. Cooperative association for internet data analysis, 2000. http: //www.caida.org/Tools/Skitter/Summary/.
2. S. Bellovin. *The icmp traceback message.* Internet Draft,IETF, March 2000. draft-bellovin-itrace-05.txt (work in progress).http://www.research.att.com/˜smb.
3. Hal Burch and Bill Cheswick. Tracing anonymous packets to their approximate source. In *Proceedings of the 14th Systems Administration Conference*, New Orleans, Louisiana, U.S.A., December 2000.
4. R.L. Carter and M.E. Crovella. Dynamic server selection using dynamic path characterization in wide-area networks. In *Proceedings of the 1997 IEEE INFOCOM Conference*, Kobe,Japan, April 1997.
5. K. Claffy and S. McCreary. Sampled measurements from june 1999 to december 1999 at the ames inter-exchange point. *Personal Communication*, January 2000.
6. Computer emergency response team. *cert advisory ca-2000-01: Denial-of-service developments*, 2000. http://www.cert.org/advisories/CA-2000-01.html.
7. Drew Dean, Matt Franklin, and Adam Stubblefield. An algebraic approach to ip traceback. In *Network and Distributed System Security Symposium, NDSS '01*, Feburary 2001.
8. W. Feller. *An Introduction to Probability Theory and Its Applications(2nd edition)*, volume 1. Wiley and Sons, 1966.
9. P. Ferguson and D. Senie. *Network ingress filtering: Defeating denial of service attacks which employ IP source address spoofing.* RFC2267,IETF, January 1998.
10. John D. Howard. *An Analysis of Security Incidents on the Internet.* PhD thesis, Carnegie Mellon University, 1998.

11. Lucent Lab. Internet mapping, 1999. http://cm.bell−labs.com/who/ches/map/dbs-/index.html.
12. Ratul Mahajan, Steven M. Bellovin, Sally Floyd, John Ioannidis, Vern Paxson, and Scott Shenker. Controlling high bandwidth aggregates in the network. Technical report, AT&T Center for Internet Research at ICSI (ACIRI) and AT&T Labs Research, February 2001.
13. K. Park and H. Lee. On the effectiveness of probabilistic packet marking for ip traceback under denial of service attack. In *Proceedings of IEEE INFOCOM 2001*, 2001.
14. Kihong Park and Heejo Lee. On the effectiveness of router-based packet filtering for distributed dos attack prevention in power-law internets. In *Proceedings of the 2001 ACM SIGCOMM Conference*, San Diego, California, U.S.A., August 2001.
15. Stefan Savage, David Wetherall, Anna Karlin, and Tom Anderson. Practical network support for ip traceback. In *Proceedings of the 2000 ACM SIGCOMM Conference*, August 2000. http://www.cs.washington.edu/homes/savage/traceback.html.
16. Alex C. Snoeren, Craig Partridge, Luis A. Sanchez, Christine E. Jones, Fabrice Tchakountio, Stephen T. Kent, and W. Timothy Strayer. Hash-based ip traceback. In *Proceedings of the 2001 ACM SIGCOMM Conference*, San Diego, California, U.S.A., August 2001.
17. Dawn X. Song and Adrian Perrig. Advanced and authenticated marking schemes for ip traceback. In *Proceedings of IEEE INFOCOM 2001*, 2001. http://paris.cs.berkeley.edu/ perrig/projects/iptraceback/tr-iptrace.ps.gz.
18. I. Stoica and H. Zhang. Providing guaranteed services without per flow management. In *Proceedings of the 1999 ACM SIGCOMM Conference*, Boston,MA, August 1999.
19. W. Theilmann and K. Rothermel. Dynamic distance maps of the internet. In *Proceedings of the 2000 IEEE INFOCOM Conference*, Tel Aviv, Israel, March 2000.
20. S. Felix Wu, Lixia Zhang, Dan Massey, and Allison Mankin. *Intension-Driven ICMP Trace-Back*. Interner Draft,IETF, February 2001. draft-wu-itrace-intension-00.txt(work in progress).

Tuning Delay Differentiation in IP Networks Using Priority Queueing Models

Pedro Sousa, Paulo Carvalho, and Vasco Freitas

Universidade do Minho, Departamento de Informática,
4710-059 Braga, Portugal
{pns,paulo,vf}@uminho.pt

Abstract. This article evaluates the use of Priority Queueing models to achieve delay differentiation in IP networks operating under the Class of Services paradigm. Three models are considered: proportional model, additive model and a novel hybrid schema based on the upper time limit model. The characteristics, behaviour and viability of these models are analysed as regards traffic delay differentiation. The impact of each model on traffic aggregation and on individual flows is also evaluated. This study is complemented by the analysis of delay differentiation from an end-to-end perspective. An adaptive differentiation mechanism is also proposed and discussed.

1 Introduction

The new Internet applications require the rethinking of network protocols. To fulfill application requirements, new philosophies oriented to Quality of Service (QoS) provision are under research and development [1]. Two architectures have been distinguished in this domain: the Integrated Services [2] and the Differentiated Services [3]. Due to its simplicity and capacity of co-existing with the actual TCP/IP protocol stack, DiffServ architecture has been pointed out as a solution to provide a limited set of QoS profiles to users.

In this work, special attention is given to a particular QoS parameter - delay - and to scheduling mechanisms which are able to obtain delay differentiation. This delay differentiation can be extremely useful to integrate real-time applications, such as voice/video and other delay-sensitive applications [4]. Even when admission control mechanisms or reservation protocols (e.g. RSVP [5]) are not present, acceptable QoS can be obtained in the presence of an appropriate delay differentiation mechanism. Additional models can be useful to provide resources to the classes/flows (e.g. bandwidth [6,7]) or in more relaxed models, to provide applications with adaptive and tolerant mechanisms [8,9,10].

This study examines the behaviour of three different delay differentiation models. The Proportional Model was considered as an efficient form to assure a proportional delay differentiation between traffic classes [11,12,13]. The Additive Model constitutes an alternative way to differentiate delays [12]. These two models are revisited and additional studies including flow granularity and

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 709–720, 2002.

end-to-end perspectives are evaluated. A more rigid schema called Upper Time Limit Model is also discussed and a novel hybrid differentiation mechanism is presented. In this context, this paper intends to pursue and complement the work presented in [11,12,13] and investigate an Upper Time Limit model. For this purpose a *Network Simulator* [14] based testbed was implemented in order to analyse the models responsiveness. The developed testbed was validated with mathematical results, including Priority Queueing Theory and Conservation Law confirmations. The three differentiation models are studied for short-time scales and for different load conditions. The goal is to understand the behaviour of each one according to the configuration parameters. Apart from the ability of each model to differentiate delay, jitter is measured for individual flows sharing a class. An adaptive differentiation mechanism is also proposed and discussed. Finally, the end-to-end relative differentiation between flows sharing a common set of differentiation nodes is investigated.

2 Proportional, Additive, and Upper Time Models

Proportional, Additive and Upper Time Limit models belong to Priority Queueing (PQ) models [15]. In PQ models each queue is ruled by a priority function that varies over time (Time-Dependent Priorities). Different models can be implemented using Time Dependent Priorities. The nature of the priority function and its configuration parameters define the behaviour of the service assigned to each queue. The following subsections review the three models briefly. The study considers N distinct classes $C_{i(0 \leq i \leq N-1)}$ having C_0 the highest priority.

2.1 Proportional Model

Let $p_i(t)$ be the priority function associated with the queue i and U_i the corresponding differentiation parameter. In the proportional model this function is given by (1), with t_0 denoting the arrival time of packet to queue i and $U_0 > U_1 > ... > U_{N-1}$. The behaviour of (1) for two packets belonging to distinct classes is depicted in Fig. 1(a). As seen U_i represents the slope of the priority function. The expected behaviour of a scheduler operating under (1) is that, under heavy load conditions, the relation (2) is valid for all classes, i.e. $0 \leq i, j < N$, where $\bar{d_i}, \bar{d_j}$ are the mean queueing delays of the classes i and j. In other words, the higher the differentiation parameter is, the lower the delay in the class will be. Furthermore, the proportional relation expected in the delays results from the proportionality in the differentiation parameters.

$$p_i(t) = (t - t_0) * U_i \qquad (1) \qquad\qquad \frac{U_i}{U_j} \approx \frac{\bar{d_j}}{\bar{d_i}} \qquad (2)$$

2.2 Additive Model

The additive model differentiates queues by an additive constant as expressed in (3), with $U_0 > U_1 > ... > U_{N-1}$. In this option, the priority difference between

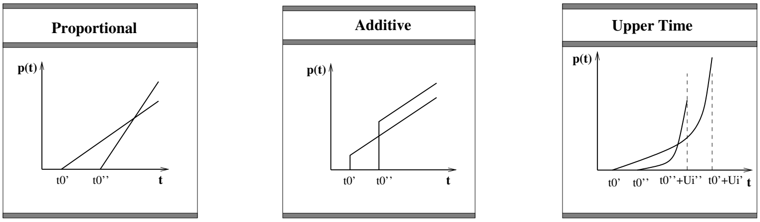


Fig. 1. (a) Proportional model (b) Additive model (c) Upper Time Limit model.

two packets remains constant over time, as depicted in Fig. 1(b). The interesting point in this model is to study the possibility of achieving additive differentiation in class delays, as expressed by (4). The equation (4) denotes that high priority classes may have a delay gain over low priority classes similar to the difference between the differentiation parameters. If this is true, it is an effective solution to spread class delays by a predefined value.

$$p_i(t) = (t - t_0) + U_i \quad (3) \qquad [\bar{d}_i - \bar{d}_j] \approx [U_j - U_i] \quad (i > j) \quad (4)$$

2.3 Upper Time Limit Model

The Upper Time Limit is a more rigid schema than additive and proportional models as it imposes a finite queueing delay. The idea is to define a boundary (reflected in U_i) for the packet queueing time (see (5)). In this model, the lower the boundary time is, the higher the priority function slope will be. At the limit $((t - t_0) \geq U_i)$ the server is *forced*[1] to dispatch the packet waiting service (see Fig. 1(c)). This model protects high priority classes, aiming that packets remain in queue for a maximum value U_i. In this model $U_0 < U_1 < ... < U_{N-1}$. Relations (2) and (4) for the Proportional and Additive models were obtained by the division and difference between the corresponding priority functions. The ratio (R) between priority functions of adjacent classes is defined[2] in (6) and represented in Fig. 2(a).

$$p_i(t) = \begin{cases} \frac{(t-t_0)}{U_i - t + t_0} \;\; if \;\; t < t_0 + U_i \\ \\ \infty \;\; if \;\; t \geq t_0 + U_i \end{cases} \quad (5) \quad R_{\frac{i}{i+1}} = \frac{p_i(t)}{p_{i+1}(t)} = \frac{\frac{t}{U_i - t}}{\frac{t}{U_{i+1} - t}} = \frac{U_{i+1} - t}{U_i - t} \; (6)$$

The function evaluates roughly in a constant proportional mode (approximately with a value of $(\frac{U_{i+1}}{U_i})$ between the classes (white area), and as the time limit arrives, the function increases (grey area) tending then to infinity (black area). Our interest in this model is to use its capabilities to limit the queuing delay on the higher priority class. This class is oriented to extreme delay-sensitive

[1] Obviously when congestion occurs, or the load of high priority classes becomes very high, packets can be dropped or the waiting time limit exceeded.

[2] For simplicity simultaneous packet arrival times are assumed, i.e. t_0 is eliminated.

applications where a bound on delay is mandatory. Our objective is to establish such delay bounds and, simultaneously, achieve proportional differentiation between the other classes. This can be obtained by combining differentiation parameters conveniently. Fig. 2(b) shows an example where $Clas_1$ is *protected* by a realistic upper time limit, and $Class_2$ and $Class_3$ with *virtual* parameters (i.e. a queueing time limit much higher than the expected for the class and $U_2, U_3 \gg U_1$). Proportionality between $Class_2$ and $Class_3$ is obtained by configuring parameters as explained in Section 2.1, considering now that higher classes have lower differentiation parameters. As shown in Fig. 2(b) there is a server working region (slashed area) where hybrid differentiation is feasible.

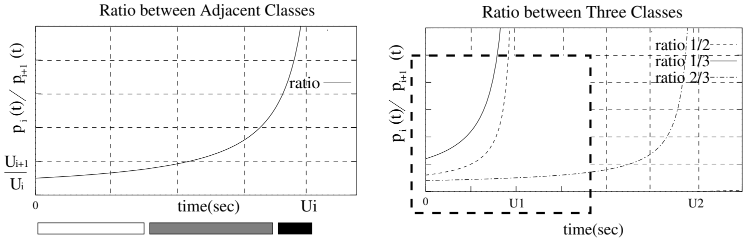


Fig. 2. (a) Ratio between priority functions (b) Expected server working region.

3 Performance Evaluation

3.1 Experimental Framework

The differentiation mechanisms were implemented and tested in the *network simulator* (NS). Each mechanism determines the scheduler behaviour. The schedulers were implemented in C++ from *Queue Class* inheritance. Proprietary queues and monitors were also developed in order to collect results from the tests. Fig. 3(a) shows the implemented architecture. At Otcl level, the user selects the scheduler, defines the differentiation parameters of the queues/classes and provides classification information, i.e. $(packet_{flowid}, queue_{id})$ pairs. At the same level, the user indicates the state information granularity to be logged during scheduling. In the architecture core, the monitor module logs state information about flows/classes periodically for subsequently analysis.

3.2 General Comments

Simulation Tests: The models were tested in several simulation scenarios for different traffic patterns. The results presented here were obtained for the simulation scenario depicted in Fig. 3(b). $Class_A$ is used for on-off traffic (the duration of *on* and *off* periods follows a Pareto distribution with shape factor of

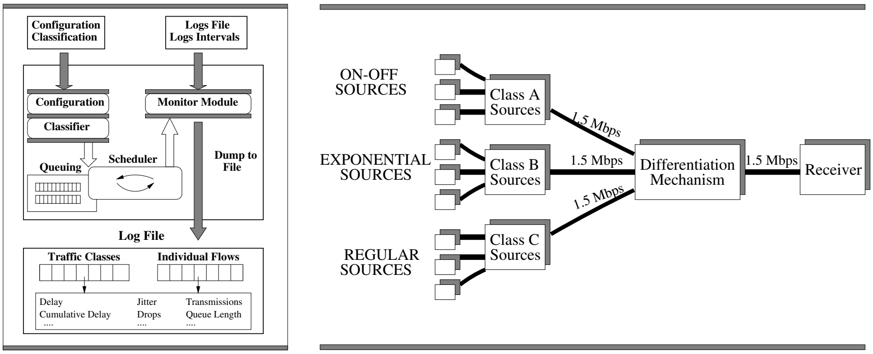


Fig. 3. (a) Developed Testbed (b) Simulation Scenario.

1.2). Additionally, $Class_B$ consists of Poisson traffic (exponential inter-arrival times), and $Class_C$ which includes regular traffic. A set of individual applications generates marked traffic for the corresponding class. The packet length is 500 *bytes*. In the delay differentiation examples presented in section 3.3, similar loads (≈33%) and queueing resources were also used for all classes. The server is connected to a 1.5 Mbps link[3]. The differentiation schemas were studied for heavy load conditions. The results are presented graphically where the *x axis* represents server packet transmission times, with a plot granularity of $80ms$.

3.3 Delay Differentiation

This section intends to illustrate the differentiation characteristics for the three models. Fig. 4 to 5 show the delay differentiation behaviour for the models. For each model, the queueing delay is plotted by sampling interval (Fig. 4 to 5(a)(c)), and its average over the simulation period (Fig. 4 to 5(b)(d)).

Proportional Model: Fig. 4(a)(b) shows the performance of the proportional model for differentiation parameters $(U_A, U_B, U_C) = (4, 2, 1)$. The results show a proportionality between the class delays. In fact, $Class_C$ (low priority class) has a delay which is on average twice the obtained by $Class_B$, which in turn has a queueing delay around two times higher than $Class_A$. This behaviour is in accordance with equation (2) assuring that for heavy load, and for acceptable configuration parameters, the proportionality relations expressed by U_i parameters generate proportionality relations between class delays.

Additive Model: Fig. 4(c)(d) illustrates the obtained results for the Additive model when $(U_A, U_B, U_C) = (0.030,\ 0.010, 0.0)$. This means that under heavy load conditions $Class_A$ may have an advantage near $20ms$ over $Class_B$. Using the same reasoning $Class_B$ may have an advantage near $10ms$ over $Class_C$ and, transitively, $Class_A$ an advantage near $30ms$ over $Class_C$. In fact, the results presented in Fig. 4(c)(d) exhibits a behaviour which verifies equation (4). This approach provides additive class delay differentiation effectively.

[3] In which there is a queue with the architecture presented in Fig. 3(a).

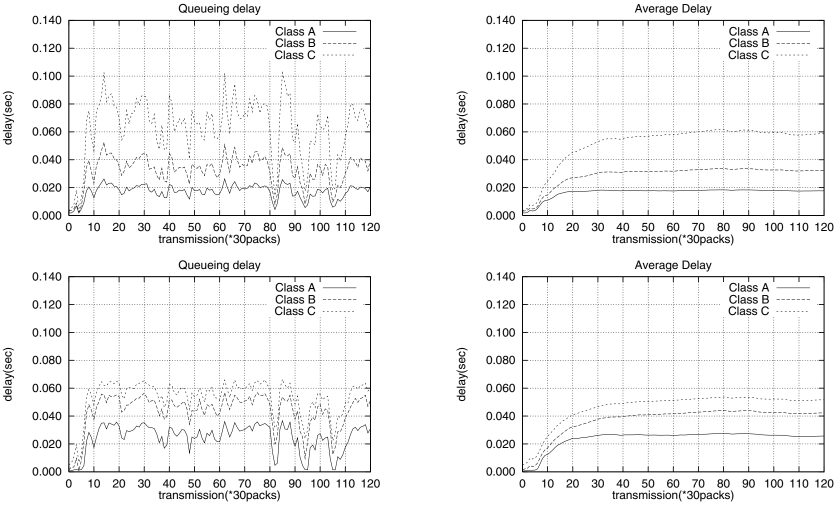


Fig. 4. (a) Queueing delay (b) Average delay for the Proportional Model, $(U_A, U_B, U_C) = (4, 2, 1)$ (c) Queueing delay (d) Average delay for the Additive Model, $(U_A, U_B, U_C) = (0.030, 0.010, 0.0)$.

Upper Time Limit Model: In this model two objectives were established: *(i)* obtain a queueing delay for $Class_A$ around $5ms$; and *(ii)* achieve proportional differentiation between $Class_B$ and $Class_C$. Fig. 5(a)(b) shows the results for this model using $(U_A, U_B, U_C) = (0.005, 0.100, 0.200)$. As explained in section 2.3 such configuration leads to a queueing delay below $5ms$ for $Class_A$ and a queueing delay for $Class_B$ two times lower than the obtained for $Class_C$. The results prove that this *hybrid* mechanism is feasible, and achieves differentiation successfully. As the results illustrate, as long as $Class_A$ queueing delays are confined to an upper-bound value ($5ms$), $Class_B$ and $Class_C$ queueing delays keep a proportional relation. Fig. 5(c)(d) represents a scenario where $Class_A$ has an upper limit of $10ms$ and $Class_C$ is assigned a queueing delay 10% higher than $Class_B$. This results in an approximation of $Class_B$ and $Class_C$ delays.

From the previous simulation examples one can argue that, under heavy load conditions, each model constitutes an effective differentiation mechanism and a good tuning scheme to provide network elements with delay differentiation capabilities. Additionally, the differentiation behaviour is achieved even in short-time scales ($80ms$ in the example). However, there are additional comments that must be made. For particular load conditions, all models can present some feasibility problems. This does not mean erroneous relative differentiation, but for particular scenarios (e.g. priority classes very loaded) the gap between queueing delays is lower than the one expressed by (2) and (4). The same occurs for the Upper Time Limit model if the traffic load on a high priority class impairs the delay limit imposed by (5). These occasional feasibility problems do not affect the essential conditions of relative differentiation, i.e. $\bar{d_0} < \bar{d_1} < \bar{d_2} < ... < \bar{d_n}$.

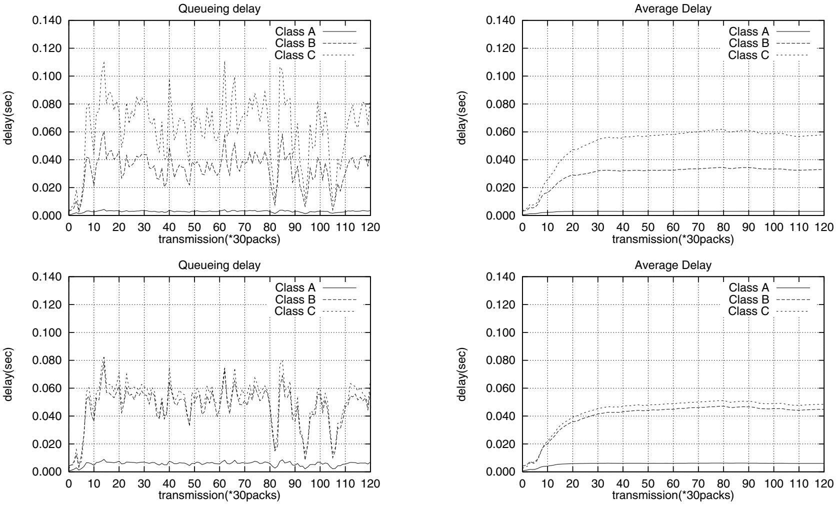


Fig. 5. (a) Queueing delay (b) Average delay for Upper Time Limit Model, $(U_A, U_B, U_C) = (0.005, 0.100, 0.200)$ (c) Queueing delay (d) Average delay for Upper Time Limit Model, $(U_A, U_B, U_C) = (0.010, 0.100, 0.110)$.

3.4 Flow Granularity

This section studies the models behaviour at flow level. The aim is to examine how the delay-oriented QoS provided to each class is extended to the flow level. The knowledge of such flow characteristics can be useful since clever applications mechanisms (e.g. adaptive) can be deployed to improve their performance.

Per Flow Queueing Delay Consistency: The differentiation mechanisms should be fair for flows sharing a class. This means that for a generic time interval $[t_0, t]$, the queueing delay associated with each class should evenly affect flows belonging to that class. To verify this, the average queueing delay for two randomly selected flows of each class on each model is plotted in Fig. 6. As shown, different priority flows share the same delay relations of the classes. Additionally, flows in the same class have identical average queueing delays. This demonstrates the fairness of differentiation mechanisms even at flow level.

Delay Variation/Jitter: Jitter is an important measure from the applications' perspective. Many real time applications adapt themselves to network conditions. For example, applications involving isochronous media need to monitor delay and jitter in order to regulate the receiver's playout buffer [16]. If an adaptive perspective is assumed [9], applications can move across different traffic classes in order to achieve more suitable delay/jitter. Additionally to average queueing delays relations among classes/flows, it is important to find foreseeable relations for jitter. The distance between the max(+)/min(-) delay lines gives an estimative of the jitter experienced by the flow. The results show that jitter either is reduced or does not change significantly when the flow moves to high

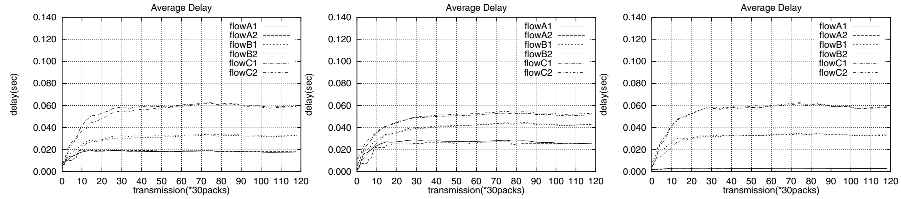


Fig. 6. Average queueing delay at Flow Level - (a) Proportional Model $(U_A, U_B, U_C) = (4, 2, 1)$, (b) Additive Model $(U_A, U_B, U_C) = (0.030, 0.010, 0.0)$, (c) Upper Time Limit Model $(U_A, U_B, U_C) = (0.005, 0.100, 0.200)$.

priority classes. Note that jitter issues are relevant for other than heavy load conditions (e.g. Fig. 7). In fact, it is possible that at t_0, when server is under heavy load conditions, $Class_i$ achieves a queueing delay $\bar{d}_i$, and subsequently at t_1, due to a load decrease on the server, the delay experienced by $Class_i$ may decrease sharply to a very low value, or even to zero. Therefore, jitter can assume a value $\bar{d}_i - 0 = \bar{d}_i$. Consequently, relations between jitter on each class are dependent on the relations between those classes' queueing delays.

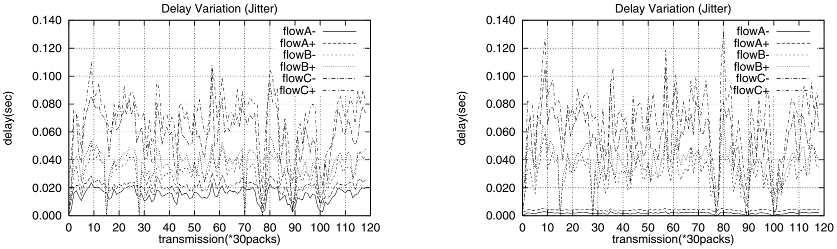


Fig. 7. Jitter at Flow Level, (a) Proportional $(U_A, U_B, U_C) = (4, 2, 1)$ (b) Upper Time $(U_A, U_B, U_C) = (0.005, 0.100, 0.200)$.

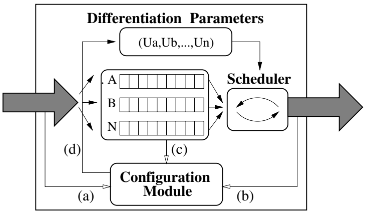
Playout Buffer dimensioning: The knowledge of jitter characteristics can be useful for playout buffer dimensioning if an adaptive behaviour of the application is assumed. When moving to low (high) priority classes, the application should protect itself by increasing (decreasing) the playout buffer. The ratio $\bar{d}_i/\bar{d}_j$ should guide the dimensioning degree adopted by the application. For the proportional model and using formula (2), a possible update strategy[4] is presented by (7), where $f_{i \Rightarrow j}$ is a flow moving from $Class_i$ to $Class_j$ and b_l the playout buffer length. Similar considerations can be made for the other models.

$$f_{i \Rightarrow j} : new(b_l) = old(b_l) * \frac{U_i}{U_j} \qquad (7)$$

[4] In this case it is assumed a single node between the sender and the receiver.

4 Adaptive Behaviour of Differentiation Mechanisms

Network elements may assume adaptive behaviours (e.g. [17] presents a dynamic regulator mechanism for real-time traffic and [18] an adaptive packet marking scheme to achieve throughput differentiation). In our opinion, this principle can be applied at scheduler level in order to improve network resources usage (e.g. [19] suggests an Adaptive-Weighted Packet scheduler for premium service). The aim is to allow scheduling to react to certain operational conditions, modifying differentiation parameters *on-the-fly*. A possible adaptive scheduler architecture, which can be easily adopted in the three models, is proposed in Fig. 8(a).



```
$A_T \Leftarrow EstimateTotalArrivalLoad()$
$A_A \Leftarrow EstimateClassArrivalLoad(A)$
$Ratio \Leftarrow \frac{A_A}{A_T}$
$if(0 \leq Ratio \leq 0.4) \Rightarrow (U_A, U_B, U_C) = (16, 4, 1)$
$if(0.4 < Ratio \leq 0.6) \Rightarrow (U_A, U_B, U_C) = (4, 2, 1)$
$else \Rightarrow \{(U_A, U_B, U_C) = (2, 1.5, 1)$
```

Fig. 8. (a) Adaptive differentiation mechanism (b) Configuration module behaviour.

The *Configuration Module* has three distinct inputs: *(a)* input traffic information (e.g. aggregated load per class), *(b)* output traffic information (level of throughput share) and *(c)* QoS node state information (e.g. queueing delay, packet loss, jitter per class). Combining this information, the *Configuration Module* can modify the differentiation parameters *(d)* to achieve a certain objective. As a simple example, consider the *Configuration Module* has having the behaviour described in Fig. 8(b). When the thresholds are violated, new differentiation parameters are evaluated in order to deny excessive throughput allocation to high priority class. As shown in Fig. 9, the node becomes reactive to $Class_A$ load limits violation. As a consequence, new parameters are assigned to each class, resulting in a delay approximation between classes. Other variants of this mechanism for the three differentiation models will be matter of further research.

5 End-to-End Relative Delay Differentiation

This section studies the proportional, additive and upper time limit models from an end-to-end perspective and establishes the corresponding upper bound limits for delay differentiation. The definition of a differentiation domain aims to achieve a foreseeable relative differentiation for flows[5] crossing a common set of

[5] As explained in subsection 3.4 the delay differentiation achieved for traffic aggregates is valid at flow level.

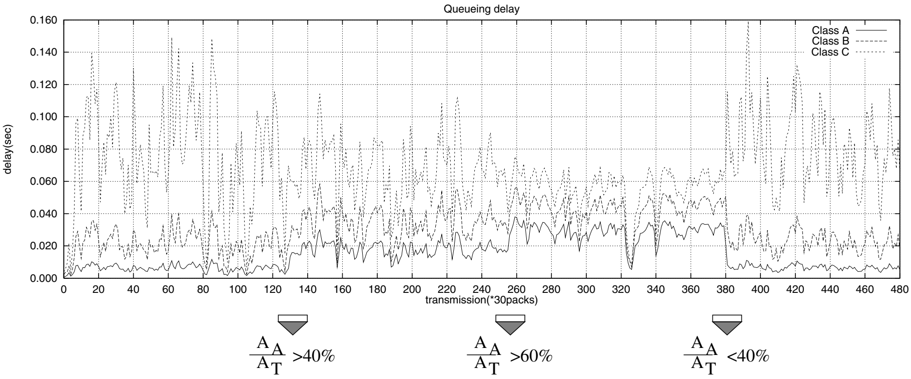


Fig. 9. Adaptive Proportional Model, (U_A, U_B, U_C)=(16,4,1)$\Longrightarrow$(4,2,1)$\Longrightarrow$(2,1.5,1).

nodes in a given time period. This domain consists of M differentiation nodes, $0 \leq j \leq M-1$, traversed by individual flows. Let $\bar{d}_i^j$ be the average queueing delay of $Class_i$ at node j. If a flow crosses M servers $(0 \leq j \leq M-1)$ then the end-to-end average queueing delay $(\bar{d}_i^*)$ of $Class_i$ can be expressed by (8). For additive differentiation and under heavy load, (9) can be applied to a generic server, where U_i^j is the differentiation parameter of $Class_i$ in node j. Considering a flow crossing M independent nodes, (9) becomes (10), which combined with (8) results in (11).

$$\bar{d}_i^* = \sum_{j=0}^{M-1} \bar{d}_i^j \tag{8}$$

$$(\bar{d}_{i+1}^j - \bar{d}_i^j) \approx (U_i^j - U_{i+1}^j) \Rightarrow \bar{d}_{i+1}^j \approx (U_i^j - U_{i+1}^j) + \bar{d}_i^j \tag{9}$$

$$\sum_{j=0}^{M-1} \bar{d}_{i+1}^j \approx \sum_{j=0}^{M-1} (U_i^j - U_{i+1}^j) + \sum_{j=0}^{M-1} \bar{d}_i^j \tag{10}$$

$$\bar{d}_{i+1}^* \approx \left[\sum_{j=0}^{M-1} (U_i^j - U_{i+1}^j) + \bar{d}_i^* \right] \tag{11}$$

Equation (11) is obtained considering all servers in the flows path under heavy load conditions and for feasible configurations, otherwise the distance between class delays can become smaller than that. Equation (12) denotes this aspect and establishes an upper-bound for the end-to-end additive differentiation behaviour between two adjacent classes. A constant ϵ is introduced due possible inaccuracies of the models when the average delays are measured in very small time scales and, simultaneously, the server is under high class load oscillations.

$$(\bar{d}_{i+1}^* - \bar{d}_i^*) < \sum_{j=0}^{M-1} (U_i^j - U_{i+1}^j) + \epsilon \tag{12}$$

For the proportional model, (9) is now replaced by (13). Considering again a generic case of M servers under heavy load conditions, equation (13) becomes (14). The right term of equation (14) can be expanded as expressed by (15).

$$\frac{\bar{d}^j_{i+1}}{\bar{d}^j_i} \approx \frac{U^j_i}{U^j_{i+1}} \Rightarrow \bar{d}^j_{i+1} \approx \left(\frac{U^j_i}{U^j_{i+1}}\right) * \bar{d}^j_i \tag{13}$$

$$\textstyle\sum_{j=0}^{M-1} \bar{d}^j_{i+1} \approx \sum_{j=0}^{M-1} \left[\left(\frac{U^j_i}{U^j_{i+1}}\right) * \bar{d}^j_i\right] \tag{14}$$

$$\frac{U^0_i}{U^0_{i+1}} * \bar{d}^0_i + \frac{U^1_i}{U^1_{i+1}} * \bar{d}^1_i + ... + \frac{U^{M-1}_i}{U^{M-1}_{i+1}} * \bar{d}^{M-1}_i \tag{15}$$

Defining now X and Y as (16), the equation (15) can be bounded by (17). Using the same arguments of the additive models and considering equations (8) (14) and (17), equation (18) gives an upper bound limit for end-to-end proportional delay differentiation between two adjacent classes.

$$X = \min_{0 \leq j \leq M-1} \left(\frac{U^j_i}{U^j_{i+1}}\right) \quad Y = \max_{0 \leq j \leq M-1} \left(\frac{U^j_i}{U^j_{i+1}}\right) \tag{16}$$

$$X * \left(\textstyle\sum_{j=0}^{M-1} \bar{d}^j_i\right) \leq (15) \leq Y * \left(\textstyle\sum_{j=0}^{M-1} \bar{d}^j_i\right) \tag{17}$$

$$\left(\frac{\bar{d}^*_{i+1}}{\bar{d}^*_i}\right) < \max_{0 \leq j \leq M-1} \left(\frac{U^j_i}{U^j_{i+1}}\right) + \epsilon \tag{18}$$

For the upper time limit model, a high priority class under feasible load conditions is expected to obtain a maximum queueing delay equivalent to the sum of the differentiation parameters associated with $Class_0$ along the path (19). Within the hybrid model presented in section 2.3, low priority classes obtain an end-to-end relation also expressed by equation (18).

$$\bar{d}^*_0 = \textstyle\sum_{j=0}^{M-1} \bar{d}^j_0 < \sum_{j=0}^{M-1} U^j_0 + \epsilon \tag{19}$$

We aim to corroborate equations (12), (18) and (19), along with ϵ significance, using simulation. The knowledge of this end-to-end differentiation behaviour is fundamental to provide applications with effective adaptation.

6 Conclusions

This work assesses the use of PQ models to achieve foreseeable delay differentiation between traffic classes. In particular, three differentiation models are studied: proportional, additive and a hybrid one using an upper time limit model. A simulation testbed has been used and validated using theoretical models. All models show acceptable consistence in achieving the expected delay differentiation behaviour, i.e., both the proportional, additive and hybrid models distribute queueing delays among traffic classes. In our opinion, they are simple and useful to tune delay differentiation in IP networks. In particular, the hybrid differentiation mechanism shows consistence in limiting queueing delay on the highest priority class and, simultaneously, achieving proportional differentiation between the other classes. An additional study at flow aggregate level was carried out focusing on flow queueing delay consistence and jitter differentiation. In order to provide scheduling elements with adaptive behaviour, an adaptive differentiation architecture is proposed. Finally, the bounds for end-to-end delay differentiation between flows crossing a relative differentiation domain were determined for each model. As future work, the tuning process, which will be further consolidated, will include other performance aspects of differentiation domains using the differentiation delay strategies considered.

References

1. G. Armitage. *Quality of Service in IP Network Foundations for a Multi-Service Internet*, Macmillan Technical Publishing, Apr. 2000.
2. R. Braden *et al. Integrated Services in the Internet Architecture: an Overview.* RFC1633, Jul. 1994.
3. S. Blake *et al. An Architecture for Differentiated Services*, RFC2475, Dec. 1998.
4. M. Baldi. *End-to-End Delay Analysis of VideoConferencing over Packet-Switched Networks*, IEEE/ACM Trans. on Net., Vol. 8, N. 4, Aug. 2000.
5. R. Braden *et al. Resource Reservation Protocol RSVP).* RFC2205, Sep. 1997.
6. I. Stoica and H. Zhang. *Providing Guaranteed Services without Per Flow Management*, In Proc. of SIGGCOMM'99, 1999.
7. Z. Zhang *et al. Decoupling QoS Control from Core Reuters: A Navel Bandwidth Broker Architecture for Scalable Support of Guaranteed Services*, In Proc. of SIGCOMM'OO, 2000.
8. P. Sousa and V. Freitas. *A framework for the development of tolerant real-time applications*, Computer Networks and ISDN Systems 30, 1531–1541, Dec. 1998.
9. T. Nandagopal, N. Venkitaraman. *Delay Differentiation and Adaptation in Core Stateless Networks*, Proc. of INFOCOM 2000, Tel Aviv, Israel - Volume 2.
10. 1. Busse et al. *Dynamic QoS Control of Multimedia Applications based on RTP*, Computer Communications, Vol. 19, N. 1, pp. 49-58, Jan. 1996.
11. C. Dovrolis and P. Ramanathan. *A Case for Relative Differentiated Services and the Proportional Differentiation Model*, IEEE Network Magaaine, 1999.
12. C. Dovrolis and D. Stiliadis. *Relative Differentiated Services in the Internet: Issues and Mechanisms*, In Proc. of ACM SIGMETRICS'99.
13. C. Dovrolis *et al. Proportional Differentiated Services: Delay Differentiation and Packet Scheduling*, In Proc. of ACM SIGCOMM'99, 1999.
14. *ns* Documentation. `http://www.isi.edu/nsnam/ns/ns-documentation.html`
15. G. Bolch *et al. Queueing Networks and Markov Chains - Modeling and Performance Evaluation with Computer Science Applications*, John Wiley & Sons INC., 1998.
16. W. E. Naylor and L. Kleinrock. *Stream Trafic Communications in Packet Switched Networks: Destination Buffering Considerations*, IEEE Trans. on Communications, VOL. COM-30, No 12, 1982.
17. S. Introu and I. Stavrakakis. *A Dynamic Regulation and Scheduling Scheme for Real-Time Traffic Management*, IEEE/ACM Trans. on Net., Vol. 8, N.l, Feb. 2000.
18. W. Feng *et al. Adaptive Packet Marking for Maintaining End-to-End Throughput in a Differentiated-Services Internet*, IEEE/ACM Trans. on Net., Vol.7, N.5, Oct. 1999.
19. H. Wang *et al. Adaptive-Weighted Packet Scheduling for Premium Service* In Proc. of the IEEE Int. Conf. on Communications, Helsinki, Finland, Jun. 2001.

QoS-Conditionalized Handoff for Mobile IPv6

Xiaoming Fu*[1], Holger Karl[1], and Cornelia Kappler[2]

[1] Telecommunication Networks Group, Technical University Berlin
[2] Information Communication Mobile, Siemens AG

Abstract. In this paper we present a scheme that enables a mobile user to perform a "QoS-conditionalized" handoff when moving to an overlapping area in Mobile IPv6. The idea is to use a QoS hop-by-hop option piggybacked in the binding messages for QoS signaling and conditionalize a handoff upon the availability of sufficient resources along the new transmission path. Our scheme builds upon the hierarchical mobile IPv6 protocol and is especially suited for micro-mobility. It also enables the mobile node to flexibly choose among a set of available access points so that the mobile node can transmit packets through a route which offers satisfying QoS.

1 Introduction

With the advent of various radio access technologies, the increasing amount of IP services over wireless as well as wired networks, and the cheap availability of IP equipment, all-IP networks will be deployed in mobile environments. As a node moves within such a network, it must be reachable and able to communicate. One solution are the Mobile IP [2] and recently the MobileIPv6 (MIPv6)[4] protocols. MIPv6 ensures correct routing of packets to a mobile node (MN) when the MN changes its point of attachment within the IPv6 network. However, supporting QoS during handoffs is still a challenging problem, e.g. due to changing routes between endpoints or varying (wireless) link characteristics when connecting to different access points.

In future all-IP networks, a multitude of different wireless technologies and service providers is likely to co-exist, and hence connections to different access points can even happen at the same time. For example, in case of a wireless access network made up of different access technologies such as UMTS and Wireless LAN, the coverage areas may overlap. With such heterogeneous access networks, the need and opportunity to select among a number of available access points arises. In particular, when a mobile node has established QoS flows, it would be desirable to perform a handoff only when the QoS of these flows can be guaranteed after the handoff as well. Therefore, a handoff should not be performed if the MN's QoS requirement is not met; yet if the QoS can be met, handoff should be performed as quickly as possible. This indicates that a handoff

* Corresponding author: Xiaoming Fu, Sekr. FT 5-2, Einsteinufer 25, 10587 Berlin, Germany, email: fu@ee.tu-berlin.de

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 721–730, 2002.

should be conditionalized upon the availability of sufficient QoS resources; also, an appropriate access router (AR) based on the QoS requirements should be selectable among a set of ARs — an example scenario would be an MN moving in an area where several radio access technologies overlap. It is the main goal of this paper to describe a handoff scheme that is in this sense QoS-conditionalized.

Furthermore, many handoffs are local in the sense that the paths from the CN to the old and new ARs only diverge before the ARs; this point of divergence is called a "switching router." In the context of QoS-conditionalized handoffs, it is desirable to restrict QoS negotiations to the path between switching router and new AR, as the path between CN and switching router remains the same. Supporting such local renegotiations depends on the mobility mechanism. The IETF Hierarchical Mobile IPv6 (HMIPv6) protocol [7] is able to optimize such local mobility, but unable to carry QoS information; hence, handoffs do not take QoS into account.

Therefore, we introduce a QoS-conditionalized handoff scheme taking advantage of HMIPv6. If all nodes along the route between the AR and the switching router are capable of fulfilling the QoS request related to the handoff, the switching router will decide to perform the actual handoff (binding entries will be modified); otherwise, the old route will still be used. The process could be iterated until the QoS requirements are met or no more ARs/routes are available.

In the rest of this paper, we first describe related work in Section 2 and then present our scheme in Section 3. Section 4 discusses some possible extensions, followed by a comparison with other approaches for QoS support in mobile IP in Section 5. Section 6 concludes the paper and outlines our future work.

2 Related Work

2.1 Mobile IP

The concept of Mobile IPv6 is based on the usage of home agent. Put briefly, a mobile node entering a foreign network obtains a local IP address, the care-of-address. This address is registered with the home agent (HA) in the home network and, when necessary, with the correspondent node. The HA can then intercept all packets destined to the MN to the CoA via IP tunneling; CNs can send packets directly. While Mobile IP ensures reachability and optimizes packet routes, it suffers from signaling load and potentially long handoff latency.

To improve on these two points, Hierarchical Mobile IPv6 (HMIPv6) [7] introduces a new entity, the Mobility Anchor Point (MAP). When a MN moves into a new MAP domain (i.e., its MAP changes), it gets 2 CoAs: a Regional CoA on the MAP's subnet (RCoA) and an on-link address (LCoA). The MN then sends a BU to the MAP specifying its RCoA in the Home Address field and its source address is LCoA, as well as requests a binding (RCoA, Home Address) from its HA and CNs. If it moves locally, only the LCoA is changed and only a registration to the MAP is needed. The MAP then acts as a proxy between the RCoA and the LCoA. Packets addressed to the RCoA are then intercepted by the MAP, encapsulated and routed to the MN. While this approach is efficient and scalable for mobility support, it is unable to provide QoS support for mobile users. In practical deployment, the MAP would usually be located in the switching router (possibly a gateway of an administrative domain); such an arrangement will be assumed in the remainder of this paper.

2.2 Mobile QoS Support

As RSVP is a well established protocol for signaling, much work has been done to address the RSVP-mobile IP interaction. An analysis of the current situation is presented in [8]. Shen et al. [6] extend RSVP to support QoS signaling in mobile IP by introducing a unified flow identifier during handoff for the interworking between RSVP and mobile IP, and taking advantages of RSVP Path/Resv two-pass procedure to setup the reservation in the new path during handoffs. However, these approaches have problems regarding scalability and signaling overhead. To overcome this, Chaskar et al. [1] introduce a new IPv6 hop-by-hop packet header option called "QoS option", composed of one or more QoS objects, to carry the QoS information for the IP flows between a MN and its CN.

This option can be included in MIPv6 registration messages, namely the Binding Update (BU) and Binding Acknowledgement (BA) messages. Since the BU is sent as soon as the data transmission from the new CoA is ready to begin, the included QoS option triggers the necessary actions to set up QoS forwarding treatment along the new path. This approach does not rely on round-trip signaling such as Path/Resv of RSVP, but rather on triggering QoS forwarding treatment along the new network path in one pass; the latency for packets to get proper QoS treatment is therefore decreased. However, it does not allow mobile users to select another AR in case of insufficient resources along the route between MN and CN (the user is not even made aware of the fact). Hence, we extend this approach to allow QoS-conditionalization and specifically base our mechanism on HMIPv6 and more details will be discussed in the forthcoming next version of the internet draft [3].

3 QoS-Conditionalized Handoff for Mobile IPv6

3.1 Overview

The all-IP network is assumed to consist of routers which may also be responsible for the management of QoS resources, in which case we call them QoS entities. For the purpose of this paper, such a QoS entity acts as a black box to which QoS requests for a certain path (e.g., from the AR to the MAP) can be sent, which checks resource availability (resources would typically include link bandwidth, buffer space in the router, CPU resources, etc.) along the particular part of the route it is responsible for, and either grants the request (and reserves the resources), denies it, or, optionally, grants a reduced version of the request. Typically, QoS entities would be located at least in the ARs and in the MAPs.

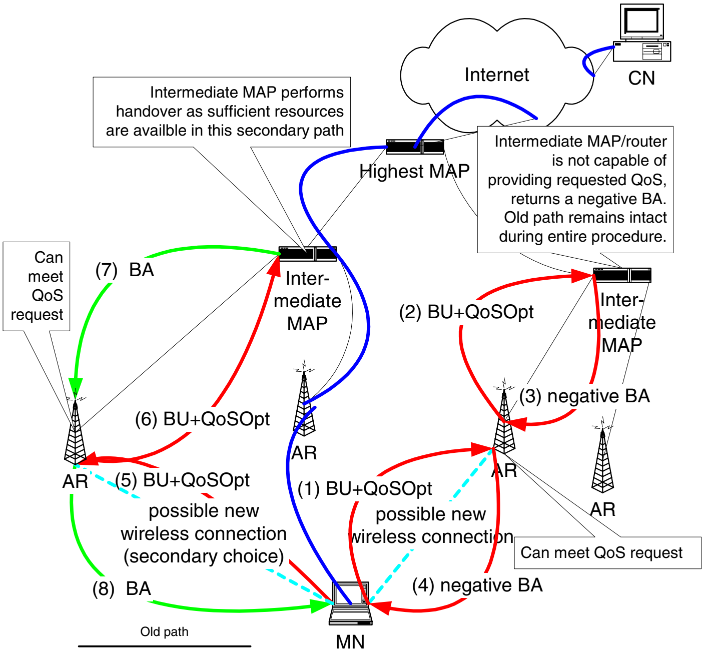

Internet
CN
Intermediate MAP performs handover as sufficient resources are availble in this secondary path
Highest MAP
Intermediate MAP/router is not capable of providing requested QoS, returns a negative BA. Old path remains intact during entire procedure.
Can meet QoS request
(7) BA
Inter-mediate MAP
(2) BU+QoSOpt
Inter-mediate MAP
(3) negative BA
(6) BU+QoSOpt
AR
AR
AR
AR
(5) BU+QoSOpt
(1) BU+QoSOpt
possible new wireless connection (secondary choice)
possible new wireless connection
Can meet QoS request
(8) BA
(4) negative BA
Old path
MN

Fig. 1. Example of the QoS-conditionalized Handoff Procedure

The operation of QoS-conditionalized handoff is as follows. A QoS hop-by-hop option is carried in the message containing the BU option to the MAP — this message is called BU+QoS message. Each QoS entity between the MN and the MAP (including the MAP) will pass the QoS requirement represented by the QoS option to internal QoS mechanisms and check its resource availability. If resources are available locally, they are reserved and the message will be forwarded along its route. If resources are not available, negative feedback will be provided to the MN by means of an extended Binding Acknowledgement (BA+QoS) message. If a BU+QoS message has reached the switching MAP and passed the local QoS test as well, the handoff will take place (the binding cache in the MAP is updated to reflect the new LCoA) and a positive BA+QoS message is returned to the MN. Otherwise, no handoff is performed and a negative BA+QoS message is returned to the MN. When observing a negative BA+QoS message, intermediate QoS entities can release reservations that could not be granted further upstream.

In order to allow both upstream and downstream QoS requirements to be considered, this approach assumes that packets for both directions follow the same route. Extending our approach to asymmetric routes should be feasible but this is beyond the scope of this paper.

Figure 1 shows one example of a QoS-conditionalized handoff procedure.

3.2 Message Format

The QoS option [1] is an IPv6 hop-by-hop header option which allows applications to specify their QoS requirements (eg., maximum/minimum bandwidth, delay) in the form of QoS objects, describing these parameters in a type-length-value format. We extend its definition by using two reserved bits in the option header: one to indicate whether only a single QoS object is present in the QoS option or if two objects are included, representing "acceptable" and "desired" level of QoS; another bit to indicate whether enough resources along the route up to the current router are available (if set it indicates failure to provide resources). These bits are called the "A" (acceptable only) and "F" (failed) bit, respectively.

3.3 Description of QoS-Conditionalized Handoff

This section describes the QoS-conditionalized handoff scheme in more detail. The main point to note is that a handoff that is local to a single MAP does not involve the CN and hence no modification to the CN is necessary; here the algorithms based on HMIPv6 in basic mode are provided.

Mobile Nodes. Algorithm 1 shows pseudo-code for the main event loop of the network-layer code of a mobile terminal. Main events to process are detecting connectivity to a new AR or loosing connectivity to an existing router and the arrival of a packet from a lower or higher protocol layer. As a simplification, the code here assumes that whenever a new AR becomes available, a handoff to this AR should be attempted; in reality, other policy-specific handoff schemes could be possible. Note that the treatment of acceptable/desirable QoS is also not shown here; the necessary modifications are reasonably straightforward.

Intermediate Nodes and MAPs. The procedure for intermediate nodes and MAPs is shown in Algorithm 2.

Note that in order to correctly process the BA+QoS message, all routers concerned with QoS management, such as MAPs, ARs, and possibly DiffServ and MPLS edge routers (ER), as well as IntServ nodes need to maintain a soft state for each flow. These states will time-out along an unused path. They can further be explicitly released via a message carrying a QoS option with "F" bit set (as illustrated in Algorithm 2) upon a successful handoff.

4 Further Discussion

4.1 Reducing the Signaling Load over the Wireless Link

As both wireless bandwidth and processing power on mobile terminals are precious resources, it would be desirable to minimize the amount of QoS information

Algorithm 1 Pseudocode of the QoS-conditionalized handoff procedure for mobile nodes

```
loop {Wait for event}
  if Event reports connectivity to an additional AR then
    MN acquires new local IP address
    Compose a BU+QoS message;
    Send BU+QoS towards MAP (via new AR);
    Increment number of ARs by 1
    if Connected = false then
      Connected $\leftarrow$ true
      Inform application of reconnection
    end if
  else if Event reports loss of connectivity to AR then
    Decrease number of ARs by 1
    if Number of ARs = 0 then
      Connected $\leftarrow$ false
      Inform application of loss of connectivity
    else if Lost AR is currently used AR then
      QoSGuaranteed $\leftarrow$ false
      Inform application of loss of QoS guarantees
    end if
  else if Event is packet from lower layer then
    if Packet is BA+QoS then
      if "F" bit not set then {request succeeded}
        Use this AR henceforth
        if QoSGuaranteed = false then
          QoSGuaranteed $\leftarrow$ true
          Inform application of QoS reestablishement
        end if
      end if{Otherwise, no action is required: either still use old AR, or QoS
      guarantees have already been lost}
    else if Packet is normal IP packet then
      Deliver to application
    end if
  else if Event is packet from higher layer then
    Forward packet, using current AR
  end if
end loop
```

traversing the wireless link and the processing in the MN. Here, Context Transfer protocol (CT) [5] appears to be particularly useful. In case CT is used, the processing in MNs and ARs must be changed accordingly. An (old) AR needs to store QoS requirement information for each of its MNs. When a MN wishes to associate itself with a new AR, it could simply inform the new AR of the old AR's identity as well as of its own address. The new AR then fetches the QoS requirement description from the old AR and initiates the BU process on behalf of the MN; BAs would still have to be provided eventually to the MN.

Algorithm 2 Pseudocode of QoS-enabled handoff procedure for routers and MAPs

```
loop {For every packet received from link layer:}
  if Packet is BU+QoS then
    if ”F” bit in BU+QoS not set then
      Ask QoS entity for resources
      if Resources are not available then
        Set “F” bit in BU+QoS packet
      end if
    end if
    if This node is a MAP then
      if This node is the switching MAP then
        Compose a BA+QoS packet from the BU+QoS packet {with “F” bit as in
        the BU+QoS packet}
        if “F” bit is not set then
          Update the MN’s binding to the new LCoA
          Optionally: Compose a negative BA+QoS message and send it along the
          old path to release reservations
        end if
        Return the BA+QoS message to the MN
      else {This node is an intermediate MAP, but not the switching MAP}
        if “F” bit is not set then
          Determine the next, hierarchically higher MAP
          Compose a new BU+QoS packet for the next MAP
          Forward this new packet to the MAP
        else {Request has failed somewhere along the path}
          Compose a BA+QoS packet (with “F” bit set)
          Return BA+QoS to the MN
        end if
      end if
    else {This node is just a normal router}
      Forward packet towards destination
    end if
  else if Packet is BA+QoS then
    if ”F” bit is set in BA+QoS then
      Release any possibly reserved resources for this flow
    end if
    Forward packet towards MN
  else {Normal packet}
    Forward packet with proper QoS treatment
  end if
end loop
```

4.2 Upgrading the Level of QoS

Another concern is which level of QoS requirements is appropriate for a MIPv6 QoS solution. As the MN requests a "(acceptable QoS, desired QoS)" pair in the new path, it can obtain any level above the acceptable QoS provided that there are sufficient resources in the path, depending on the policy of the provider. To reduce the difficulty in authorization/charging (which may be based on previously used QoS), we assume that the route between the switching router and the CN has already been guarantteed with the desired QoS level, hence only the remaining part (mobile part, typically in an access network) will be actually effected.

4.3 Macro-Mobility Consideration

Our scheme is mainly designed for intra-MAP mobility cases but it can be extended for macro-mobility scenarios. For handoffs between different domains (i.e., there is no MAP on the joint part of the paths CN ↔ old AR and CN ↔ new AR), the BU+QoS message could be sent to the CNs directly after an MN moves to a new AR. All the intermediate routers follow the procedure outlined in Algorithm 2 as indicated in Section 3.3 and the CN now takes, in a sense, the responsibility of a MAP. However, the mechanisms to detect an inter-MAP mobility and switch between an intra-MAP mobility and an inter-MAP mobility for both CN and MN require further study.

5 Comparison with Other Proposals

This section compares the proposed scheme with two other proposals to support QoS in mobile IP: QoS framework for Mobile IPv6 [1] and RSVP for MIPv6 [6].

First, the latency to re-establish QoS forwarding mechanisms is vital for a QoS solution in mobile IP. As an example, suppose a handoff is local and the transmission delay from a MN to MAP is 10 ms, considerately lower than the end-to-end delay from MN to CN, 100 ms. Then our QoS-enabled handoff scheme needs a period of two passes for signaling from MN to switching router, which is 20 ms, for both downstream and upstream QoS re-establishment, and actual downstream forwarding can take place already after 10 ms (once the MAP has received the BU+QoS, downstream forwarding can already begin). The approach in [1] needs an even lower latency for upstream QoS re-establishment (10 ms), but only if the resources are actually available. For downstream QoS re-establishments, [1] needs two passes for signaling delay from MN to CN (or a local mobility agent if micro-mobility solution is used), 200 ms (or 20 ms). [6] needs two and three passes signaling delay from MN to MAP for upstream and downstream QoS re-establishment, 20 ms and 30 ms, respectively. It shows our scheme performs relatively well in the local mobility case. Choosing another AR when one route is unable to accommodate the QoS request is possible with our approach and [6], yet not possible using [1]. Finally, all three approaches compared enhance the efficiency of QoS signaling for handoff, but our approach is the only one that can ensure that a handoff is performed only when QoS requirements are met. A summary of the comparison is given in Table 1.

Table 1. Comparison with other protocols

		QoS-Conditionalized Handoff	QoS Framework for MIPv6	RSVP for MIPv6
Latency for QoS re-establishment	Down-stream	Two passes from MN to switching router	Two passes from MN to CN or local agent	Two passes from MN to switching router
	Upstream	Two passes from MN to switching router	One pass from MN to switching router	Three passes from MN to switching router
Ability to choose another AR		Yes	No	Possible
Ability to perform QoS-conditionalized handoff		Yes	No	No

6 Conclusions and Future Work

QoS support in all-IP mobile networks brings about great challenges and requirements. This paper presents a hierarchical, flexible, and scalable solution that makes use of an IPv6 hop-by-hop option. Our scheme reduces the signaling bandwidth on the backbone by hiding local mobility while still providing ability to do QoS signaling. Our work extends the work in [1] by: 1) enabling mobile users to choose a "good" access point when several (or overlapping) ones are available (e.g., WLAN and UMTS in hot spots); 2) having handoffs QoS-conditionalized, i.e., handoffs could be performed only when QoS requirements are met or most satisfied. The latency for QoS re-establishment is reduced compared to RSVP-based approaches during a handoff.

We are extending the work presented in this paper as follows: 1) Prototype implementation of QoS option and measuring the benefits of applying our scheme; 2) the QoS option may be changed or misused by attackers, hence we also study how to appropriately secure the QoS-conditionalized handoff procedure.

Other future research items include incorporating our scheme with other mobility solutions such as fast handoff and experimenting with adaptive applications using our scheme.

Acknowledgement. The authors are very grateful to Andreas Festag, Günter Schäfer, Changpeng Fan, and Mirko Schramm, who have contributed many important ideas as well as to Hemant Chaskar and Hesham Soliman for their insightful suggestions to an earlier version of this paper.

References

1. H. Chaskar and R. Koodli. A Framework for QoS Support in Mobile IPv6. Internet Draft draft-chaskar-mobileip-qos-Ol.txt (work in progress), March 2001.
2. C. Perkins (ed.). IP Mobility Support. RFC 2002, October 1996.
3. A. Festag, X. Fu, H. Karl, G. Schäfer, C. Fan, C. Kappler, and M. Sehramm. QoS-Conditionalized Binding Update in Mobile IPv6. Internet draft draft-tkn-monileip-qosbinding-v6-OO.txt (work in progress), July 2001.
4. D. Johnson and C. Perkins. Mobility Support in IPv6. Internet Draft draft-ietf-mobileip-ipv6-14.txt (work in progress), July 2001.
5. R. Koodli and C. Perkins. Context Transfer Framework for Seamless Mobility. Internet Draft draft-koodli-seamboby-ctvv6-OO.txt (work in progress), February 2001.
6. C. Shen, W. Seah, A. Lo, H. Zheng, and M. Greis. An Interoperation Framework for Using RSVP in Mobile IPv6 Networks. Internet Draft (work in progress), July 2001.
7. H. Soliman, C. Castelluccia, K. El-Malki, and L. Bellier. Hierarchical MIPv6 mobility management. Internet Draft draft-ietf-mobileip-hmipv6-04.txt (work in progress), July 2001.
8. M. Thomas. Analysis of Mobile IP and RSVP Interactions. Internet Draft draft-thomas- seamoby-rsvp-analysis-OO.txt (work in progress), February 2001.

On Loss Probabilities in Presence of Redundant Packets with Random Drop

Parijat Dube[1], Omar Ait-Hellal[2], and Eitan Altman[1,3]

[1] INRIA, B.P.93, 06902 Sophia-Antipolis, France.{pdube,altman}@sophia.inria.fr
[2] Xbind Inc., 55 Broad Street, NY, NY 10004, U.S.A. oaithel@xbind.com
[3] C.E.S.I.M.O., Facultad de Ingeneria, Universidad de Los Andes, Mérida, Venezuela

Abstract. We study the loss probabilities of messages in an M/M/1/K queueing system where in addition to losses due to buffer overflow there are random losses on the incoming and outgoing links. We focus on the influence of adding redundant packets to the messages. We obtain analytical results that allow us to investigate when does adding redundancy decrease the loss probabilities.

1 Introduction

Loss rate of packets is an important performance measure in telecommunication networks. Rapid progress in the development of fiber optics allows to achieve a bit error rate of 10^{-14}; information loss is then essentially due to congested nodes and buffer overflow. However, in wireless networks random losses also occur in the channels/links apart from congestion losses. Often, when messages are divided into several packets, the loss of a packet results in the loss of the whole message. In order to reduce the losses, one may add redundant packets so that lost packets can often be reconstructed. Indeed, there exist erasure recovery codes that, by adding k redundant packets to a message, enable to reconstruct up to k losses, see e.g.[4],[6], [8]. Note, however, that by adding redundant packets, the workload increases and thus the loss probability of a packet may increase [1]. Alternatively, if one wishes to have the workload unchanged, this means that the throughput of useful information transmitted by the source decreases. Thus there are two types of tradeoffs to be studied (according to whether we want to keep the total transmitted throughput the same, or only the throughput corresponding to useful transmitted information). In this paper we are concerned with studying the loss probabilities of messages in queueing systems where in addition to losses due to buffer overflow there are also random losses on the incoming and outgoing links to the bottleneck node. In particular, we study the tradeoffs mentioned in the previous paragraph.

The problem of analyzing loss probabilities due to congestion losses in the presence of redundant packets has been addressed in several papers in the past [1,6,4,3,8]. In [6], the authors have used an approximation based on an assumption of independence between consecutive losses, and shown that redundancy results in decrease of loss rate by a factor of 10 to 100. Exact numerical methods based on recursions [4] led to an opposite conclusion, i.e. that redundancy

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 731–744, 2002.

causes increase in loss probabilities. Explicit expressions for the losses have then been developed in [3,8] and references therein which allowed to obtain regions of parameters in which Forward Error Correction (FEC) is useful and others where it is not. In particular, in [3] information theoretical type of channel capacity has been obtained for channels with congestion losses (and general service and inter-arrival times). All these references studied models of where losses is only due to congestion. Such models are useful in fiber-optic networks, when the main source of losses in the network is indeed overflow of a bottleneck buffer. There are however other situations in which a non-negligible amount of losses may also occur at noisy links.

The goal of this paper is to determine the role of redundant packets in networks in which losses may be due to both phenomena: link *random losses* and losses due to *congestion losses*. We obtain expressions that permit us to study two scenarios for adding FEC. In the first, the global transmission rate is unchanged; when adding FEC we reduce the rate of useful information. We then analyze how does the received rate of useful information depend on the FEC. In the second scenario we keep the rate of useful information unchanged; adding FEC then increases the congestion and hence the losses, but allows one to recover some losses.

The paper is structured as follows. Section 2 presents the model and motivation. Section 3 presents our main results derived using an algebraic approach involving multidimensional generating functions. Section 4 provides numerical examples and discusses the region where adding redundancy improves performance. In Sec. 5 we employ a combinatorial approach using Ballot theorems to obtain explicit expressions for loss probabilities employing techniques developed in [8]. Section 6 concludes the paper.

2 The Model and Its Motivation

We consider networks consisting of a buffer that is in-between two noisy links. The latter is a suitable model for satellite connections in which there is a noisy uplink and a noisy downlink connection with further losses that may be due to congestion inside the satellite. We assume throughout that a packet that is corrupted before it arrives to the bottleneck queue is discarded and does not occupy any buffer space. In the analysis below we shall model random losses in the incoming link (uplink) and congestion losses at the node. We consider an M/M/1 queue with a finite buffer of size K (including the packet in service). We assume that losses can be caused either by a buffer overflow or randomly with probability r. The arrival process from the source is assumed to be Poisson with rate λ and the service times of packets is exponentially distributed with rate μ. Hence, the effective arrival process to the system (buffer) can be assumed to be Poisson with rate $\lambda_e = (1-r)\lambda$. Define $\bar{r} = 1-r$, $\rho = \lambda_e/\mu$, and $\rho_r = \rho/\bar{r}$. We present a recursive scheme for computing $P(j,n)$ which is the probability of j losses (including random losses in the incoming link and congestion losses at the node) among n consecutive packets.

Remark 1. The case when there are losses in both the incoming and outgoing links can be analysed once we have $P(j,n)$. For example, let the random loss probability in the outgoing link be u and let $\mathcal{P}_{j,n}$ be the probability of j losses among n consecutive packets of a message when there are random losses with probability r in the incoming link, congestion losses due to buffer overflow at the node and random losses with probability u in the outgoing link. Then $\mathcal{P}_{j,n} = \sum_{w=0}^{j} \binom{n-j+w}{w} u^w (1-u)^{n-j} P(j-w,n)$.

Thus knowing $P(j,n)$, which is the loss probability in the model we consider (i.e., random losses in the incoming link and congestion losses at the node) one can obtain the loss probabilities for the case when random losses can occur both in the incoming and the outgoing links.

3 Approach Using Generating Functions: Main Results

For the system with Poisson arrivals with rate λ_e and exponential transmission rate μ, in steady state, the probability of finding i packets in the system at an arbitrary epoch is given by $\Pi(i) = \rho^i / \sum_{l=0}^{K} \rho^l$. Define $Q_i(k)$ to be the probability that k packets out of i leave the system during an inter-arrival epoch. We have

$$Q_i(k) = \rho\alpha^{k+1}, \quad 0 \le k \le i-1, \qquad Q_i(i) = \alpha^i, \quad \text{where } \alpha := (1+\rho)^{-1}.$$

Denote by $P_i^a(j,n)$ the probability of j losses in a block of n consecutive packets, given that there are i packets in the system just before the arrival of the first packet in the block. Since the first packet in the block is arbitrary, we have

$$P(j,n) = \sum_{i=0}^{K} \Pi(i) P_i^a(j,n). \tag{1}$$

The recursive scheme for computing $P_i^a(j,n)$ is then for $i = 0, 1, ..., K-1$:

$$P_i^a(j,1) = \begin{cases} \bar{r} & j=0 \\ r & j=1 \\ 0 & j \ge 2, \end{cases}, \text{ and } P_K^a(j,1) = \begin{cases} 1 & j=1 \\ 0 & j=0, j \ge 2. \end{cases} \tag{2}$$

For $n \ge 2$ we have for $0 \le i \le K-1$

$$P_i^a(j,n) = \bar{r} \sum_{k=0}^{i+1} Q_{i+1}(k) P_{i+1-k}^a(j,n-1) + r \sum_{k=0}^{i} Q_i(k) P_{i-k}^a(j-1,n-1),$$

$$\text{and} \qquad P_K^a(j,n) = \sum_{k=0}^{K} Q_K(k) P_{K-k}^a(j-1,n-1).$$

Next, we state the main results, whose detailed proofs are given in the Appendix. Define $q(y,z) \overset{\Delta}{=} \sum_{j=0}^{\infty} \sum_{n=1}^{\infty} y^j z^{n-1} P(j,n)$. Let $x_1(y,z)$ and $x_2(y,z)$ be the solutions in x of $x^2 - (1+\rho - r\rho yz)x + \bar{r}\rho z) = 0$:

$$x_1(y,z) = \frac{1+\rho - r\rho yz + \sqrt{(1+\rho - r\rho yz)^2 - 4\bar{r}\rho z}}{2}$$
$$x_2(y,z) = \frac{1+\rho - r\rho yz - \sqrt{(1+\rho - r\rho yz)^2 - 4\bar{r}\rho z}}{2}.$$

We shall often write simply x_1 and x_2 for $x_1(y,z)$ and $x_2(y,z)$. Define, for all $k \geq 1$, $\delta_k = x_1^k - x_2^k$, $\phi_k = (\bar{r} + ry)z\delta_{k-1} - \delta_k$. Let $R_K = (\sum_{l=0}^{K} \rho^l)^{-1}$.

Proposition 1. *The probability generating function (PGF) q is given by*

$$q(y,z) = \frac{R_K}{1 - (\bar{r} + r\rho y)z} \Big[(\bar{r} + ry)R_{K-1}^{-1} + y\rho^K + z\rho(\alpha\rho)^K(\bar{r}(y-\alpha) - \alpha\rho y)A(y,z) + rzy(\alpha\rho)^K B(y,z) \Big], \qquad (3)$$

where $A(y,z)$ and $B(y,z)$ solve

$$\begin{pmatrix} z\rho\alpha(\alpha x_1)^{K+1}(y(\bar{r} - \alpha x_1) - \bar{r}\alpha) & z\alpha^2(\bar{r}(x_1 - \rho) + r x_1 y(\alpha x_1)^K) \\ z\rho\alpha(\alpha x_2)^{K+1}(y(\bar{r} - \alpha x_2) - \bar{r}\alpha) & z\alpha^2(\bar{r}(x_2 - \rho) + r x_2 y(\alpha x_2)^K) \end{pmatrix} \begin{pmatrix} A(y,z) \\ B(y,z) \end{pmatrix}$$
$$= (-1) \begin{pmatrix} (1-\alpha x_1)\alpha x_1^{K+1} y + (1-\alpha x_1)\alpha x_1(ry + \bar{r}) \left(\frac{1-x_1^K}{1-x_1}\right) \\ (1-\alpha x_2)\alpha x_2^{K+1} y + (1-\alpha x_2)\alpha x_2(ry + \bar{r}) \left(\frac{1-x_2^K}{1-x_2}\right) \end{pmatrix}. \qquad (4)$$

For $y = 0$, Prop. 1 simplifies to: $q(0,z) = \bar{r}\left[R_{K+1}^{-1} - z\rho^K A(0,z)\right](R_K 1 - \bar{r}z)^{-1}$.

Having obtained the PGF, the explicit expressions for the required probabilities can be obtained by inverting $q(y,z)$. We next focus on $P_\rho(> j, n)$, the probability of losing more than j packets out of n. We investigate the cases of $j = 0, 1$, in order to be able to decide whether adding a redundant packet to each message results in a decrease of the loss probability. The proofs can be found in [2]. To stress the dependence of the different quantities (such as the p.g.f. q) on the random loss parameters, we shall sometimes add r and λ explicitly to the notation as subscript (e.g. we shall write $q_r^\lambda(y,z)$).

Corollary 1. *(i) $q_r^\lambda(0,z) = q_0^{\bar{r}\lambda}(0,\bar{r}z)\bar{r}$, (ii) $P_r^\lambda(0,n) = \bar{r}^n P_0^{\bar{r}\lambda}(0,n)$.*

Corollary 2. *The probability of losing one packet out of n consecutive packets, i.e., $P(1,n)$ is given by*

$$P(1,n) = [z^{n-1}] \left. \frac{\partial q(y,z)}{\partial y} \right|_{y=0} = [z^{n-1}]F_1(z) + [z^{n-1}]F_2(z)$$

$$\text{with } F_1(z) = \frac{R_K}{1-\bar{r}z}\bar{r}\left[R_{K-1}^{-1} - z(\alpha\rho)^{K+1}A(0,z)\right]\left(-1 + \frac{zr\rho}{1-\bar{r}z}\right)$$

$$F_2(y) = \frac{R_K}{1-\bar{r}z}\left[R_{K-1}^{-1} + \rho^K - z(\alpha\rho)^{K+1}\bar{r}\dot{A}(0,z) + rz(\alpha\rho)^K B(0,z)\right]$$

where $A(0,z)$ and $B(0,z)$ are values at $y = 0$ of $A(y,z)$ and $B(y,z)$ defined in Proposition 1 and $\dot{A}(0,z)$ is the derivative of $A(y,z)$ with respect to y, evaluated at $y = 0$.

4 Numerical Examples

In this section we compare the loss probabilities of a whole group of n consecutive packets, which we call a block, with and without j additional redundant packets. The group of packets that include the original block plus the additional redundant packets (if these are added) is called a frame. If at least n packets out of these consecutive $n + j$ packets reach the destination then no loss of frame occurs. In this section we restrict ourselves to the case of $j = 0$, i.e., no redundancy and $j = 1$, one redundant packet per n packets. Without loss of generality, we may scale the time so that the service rate is unity: $\mu = 1$. In the numerical examples we are looking only at the random losses in the incoming link with probability r and congestion losses. We take $K = 25$. When we numerically compared $P_\rho(> 0, n)$ with $P_\rho(> 1, n + 1)$ we always obtained $P_\rho(> 1, n + 1) < P_\rho(> 0, n)$, which should be of no surprise: this observation means that if redundancy is added in such a way that *the total load on the system remains unchanged* then indeed redundancy improves performance in terms of loss probabilities. However, the assumption that the total load remains the same means that the throughput of the *useful* information decreases (in real time applications this would mean that a higher compression rate should be used before transmission). This type of comparison (keeping the total load unchanged) has not been performed previously in [6,4,3,8] even for the case of congestion losses only. E.g., if we add k redundant packets to n (which gives frames of $n + k$) and if the load is unchanged, then this means that the throughput of useful information carried by a frame has decreased by a factor of $n/(n + k)$. Yet we have less losses of packets. Thus the question that needs to be addressed is whether we gain in *goodput* in this case. Let us define the goodput as the throughput arriving well to the destination. Then this is given by

$$\text{(input rate of blocks)} \times n/(n + k) \times P_\rho(\leq k, n + k).$$

So a meaningful thing to compare is $P_\rho(0, n)$ with $\frac{n}{n+1} P_\rho(\leq 1, n + 1)$ for fixed λ. In Fig. 1, we plot the relative gain, i.e.,

$$\frac{\frac{n}{n+1} P(\leq 1, n + 1) - P(0, n)}{P(0, n)}. \tag{5}$$

From Fig. (1) we observe that the benefits of adding FEC grows as the amount of random losses increases, and also as n increases. Also for very low r (very close to 0) and very low n (as compared to buffer size) we loose by adding FEC. Fig. (2) plots the same curve for $\lambda = 0.99$. We observe that curves for $\lambda = 0.3$ and $\lambda = 0.99$ are identical $r \geq 0.1$ and larger n and for r close to 0 the difference is very small. *Remark:* Consider a scenario in which there are only random losses (with probability r) and no congestion losses. Then we have: $P_\rho(0, n) = (1 - r)^n, \quad P_\rho(1, n) = nr(1 - r)^{n-1}$. If we want to study the effect of adding FEC on recovering from different type of losses we can compare the relative gain defined in (5) for the cases when $r = 0$ (congestion losses but no random losses) to the case when there are no congestion losses but only random

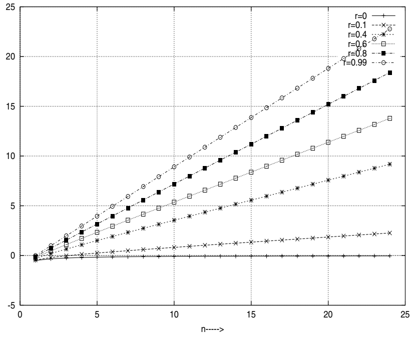


Fig. 1. $\dfrac{\frac{n}{n+1}P(\leq 1, n+1) - P(0,n)}{P(0,n)}$ as a function of n for varying r with $\lambda = 0.3$

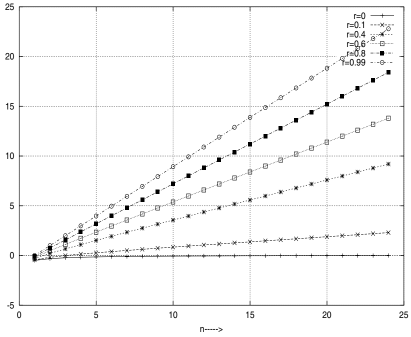


Fig. 2. $\dfrac{\frac{n}{n+1}P(\leq 1, n+1) - P(0,n)}{P(0,n)}$ as a function of n for varying r with $\lambda = 0.99$

losses with loss probabilities $P_\rho(0,n)$ and $P_\rho(1,n)$. We plot this comparison in Fig. (3) and observe that FEC is more helpful in recovering from random losses than congestion losses.

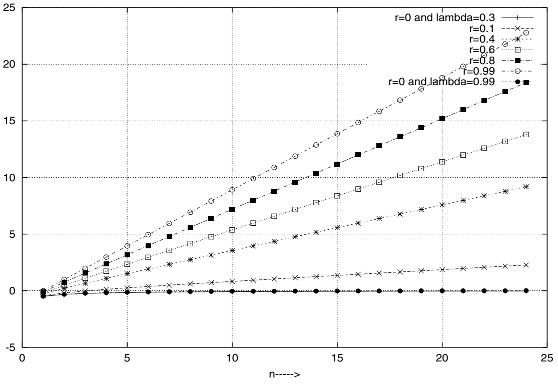


Fig. 3. Gain $\frac{\frac{n}{n+1}P(\leq 1, n+1) - P(0,n)}{P(0,n)}$ as a function of n for r varying from 0.1 to 0.99 for the scenario when there are no congestion losses. Also shown is the gain when there are no random losses ($r = 0$) and only congestion losses with $\lambda = 0.3$ and $\lambda = 0.99$. Observe that the curves for $r = 0$ and $\lambda = 0.3$ and $\lambda = 0.99$ have negligible differences.

Next we look at the case where the transmission of useful information is kept unchanged when adding redundancy. This implies that the total packet arrival rate increases due to adding redundancy. We assume that the rate at which frames arrive is the same for the two cases and is given by x. In case of no redundancy the rate at which packets arrive is $\lambda = \rho = nx$ and in case of redundancy $\lambda = \rho = (n + 1)x$. A frame is lost in the latter case if more than one packet is lost out of $n+1$ consecutive packets. We are thus interested in the difference $D = P_{nx}(>0,n) - P_{(n+1)x}(>1,n+1)$. If $D > 0$ then the redundancy decreases the loss probability of messages. Observe that

$$\begin{aligned} D &= 1 - P_{nx}(0,n) - \left[1 - P_{(n+1)x}(0,n+1) - P_{(n+1)x}(1,n+1)\right] \\ &= P_{(n+1)x}(1,n+1) + P_{(n+1)x}(0,n+1) - P_{nx}(0,n). \end{aligned} \tag{6}$$

We next plot the relative gain $\frac{D}{P_{nx}(>0,n)}$ as a function of n for $x = 0.03$ (this means the load nx, varies from 0.03 (for $n = 1$) to 0.75 (for $n = 25$)) in Figure 4 and for $x = 0.4$ (load varying from 0.4 to 10) in Figure 5. The curves show that for fixed r, there exists a value of the frame size at which the gain obtained by adding FEC as defined in (6) is maximum. These figures can thus be used in order to optimize the size of blocks to which we add FEC.

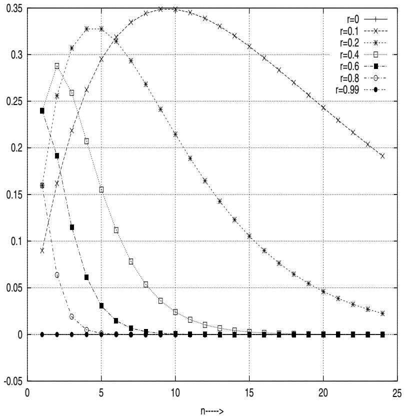


Fig. 4. $\frac{D}{P_{nx}(>0,n)}$ as a function of n for different r and $x = 0.03$. Observe that the load changes with n also.

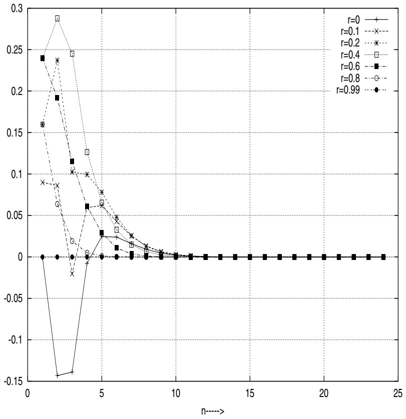


Fig. 5. $\frac{D}{P_{nx}(>0,n)}$ as a function of n for different r and $x = 0.4$

All the above curves establish that we benefit from adding redundancy when r is not very small, and this is a valid remark or observation at any load. However when the random loss probability is very low (close to 0) we may loose by adding redundancy.

5 Combinatorial Approach Using Ballot Theorems

We next employ combinatorial arguments together with the Ballot theorems [5] to alternatively obtain explicit expressions for all the probabilities of the previous section. In particular, we shall find the probability $P_i^a(j,n)$.

Consider the case when j [1] losses consist of $j_r(0 \leq j_r \leq j)$ random losses and $j_c(0 \leq j_c \leq j)$ congestion losses. The number of ways such an event can occur is $\binom{j}{j_c}$. We calculate the probability of one such outcome. The probability depends on the position of the lost packets in the frame. Let us denote by r_i the position of the ith random loss, $1 \leq i \leq j_r$ in the original frame. Also $i \leq r_i \leq n-(j_r-i)$. Thus $r_1 = 1$, when the first packet was lost by random loss and $r_{j_r} = n$, when the last packet was lost by random loss.

The following analysis is for the case of $j_r \geq 2, r_1 \neq 1 r_{j_r} \neq n$. We shall supplement the discussion with other cases $j_r \leq 1$ *and/or* $r_1 = 1$ *and/or* $r_{j_r} = n$ at appropriate places. Observe that the random losses can be *isolated* or they can occur in burst. In fact since our message length is finite (n), the probability that all the random losses occur in a burst is > 0 [2]. Also observe that only the packets of the original message which are not subject to random losses have the *potential* of getting lost at the queue due to buffer overflow (as these are the only packets that actually reach the queue). Thus we shall look at the packets of the original message between consecutive *random loss events.* A random loss event is formed consecutive random losses. Say that the packets coming to the queue between consecutive random loss events are forming an *interval.* Let T be the number of such intervals and k_i $(1 \leq i \leq T)$ be the number of consecutive random losses in the random loss event starting after the end of the ith interval and prior to the beginning of the $i+1$th interval. Thus the maximum value of T is $j_r + 1$ when all the random losses occur isolated and on the other extreme, the minimum value of T is 2 when all the random losses occur in a burst. Define $z(t) := \sum_{h=1}^{t} k_h$. We now distribute the j_c congestion losses in the T intervals of lengths $r_1 - 1, r_{1+k_1} - r_{k_1} - 1, r_{1+k_1+k_2} - r_{k_1+k_2} - 1, \ldots, r_{1+z(T)} - r_{z(T)} - 1$. Let n_y be the number of congestion losses in the yth such interval. Observe that (for $2 \leq y \leq j_r + 1$) we have $0 \leq n_y \leq \min(r_{1+z(y-1)} - r_{z(y-1)} - 1, j_c)$, and for $y = 1$, $0 \leq n_y \leq \min(r_1 - 1, j_c)$. Also, n_y satisfy $\sum_{y=1}^{T} n_y = j_c$. Now the number of ways in which n_y losses can occur in the yth interval is

$$\binom{r_{1+z(y-1)} - r_{z(y-1)} - 1}{n_y}$$

for $2 \leq y \leq T$ and is $\binom{r_1 - 1}{n_1}$ for $y = 1$.

We shall calculate the probability of one such event. We shall look at three types of intervals: A-starts with the first arrival after a random loss and ends with the last arrival before a random loss event; B-starts with the arrival of the first packet of the message (if $r_1 \neq 1$) and ends with the last arrival before the

[1] Observe that here we are looking at the case when the random losses (if any) occur before the frame enters the buffer. The complementary case of random losses occurring after the frame leaves the node can be handled as discussed in Remark 1. And then one can obtain the loss probabilities for the case when random losses can occur both in the outgoing and in the incoming link.

[2] Although bursty loss occurrence is more a characteristic of congestion losses.

first random loss event; C-starts with the first arrival after the last random loss event and ends with the arrival of the last packet of the message.

In a sample path with $j_r \geq 2, r_1 \neq 1, r_{j_r} \neq n$, and with A_i an interval of type A, the order of occurrence of the intervals is $B \rightarrow A_1 \rightarrow A_2 \ldots \rightarrow A_{T-2} \rightarrow C$. For $j_r \geq 2, r_1 = 1, r_{j_r} \neq n$, the order is $A_1 \rightarrow A_2 \ldots \rightarrow A_{T-1} \rightarrow C$ and no interval of type B. For $j_r \geq 2, r_1 \neq 1, r_{j_r} = n$, the order is $B \rightarrow A_1 \ldots A_{T-1}$ and no interval of type C. Similarly, for $j_r \geq 2, r_1 = 1, r_{j_r} = n$, there will be no interval of type either B or of type C. For $j_r = 1$, there can either be intervals $B \rightarrow C$ or C or B and no interval of type A can occur.

Let the queue length at the beginning of the yth interval be α and at the end of the interval be β. We thus need to calculate the probability of a path that starts with α packets in the buffer, ends with β packets in the buffer, has n_y losses in it by congestion and consists of $a_y = (r_{1+z(y-1)} - r_{z(y-1)} - 1)$ arrival events. We employ the arguments as in [8] to evaluate this probability. However here in our analysis we also need to know the queue length at the arrival of the last packet of an interval. We shall denote this probability by $P_{(\alpha,\beta)}(n_y, a_y)$. Let f_j denote the jth lost packet. We shall decompose an interval into three types of events as follows: (i) $\mathcal{V}_\alpha(f_1)$-the first packet to be lost is f_1 given that upon the arrival of the first packet of the interval there are α packets in the buffer; (ii)$\mathcal{S}(f_l, f_{l+1})$-packet f_{l+1} is lost given that packet f_l was lost; (iii)$\mathcal{U}(f_{n_y}, \beta)$-packet f_{n_y} is the last to be lost and the queue length at the arrival of the last packet of the interval is β.

Observe that an interval consists of the succession of events $\mathcal{V}_\alpha(f_1), \mathcal{S}(f_1, f_2)$, $\mathcal{S}(f_2, f_3), \ldots, \mathcal{S}(f_{n_y-1}, f_{n_y}), \mathcal{U}(f_{n_y}, \beta)$. Let $v_\alpha(f_1)$, $s(f_l, f_{l+1})$ and $u(f_{n_y}, \beta)$ be the probabilities of the event $\mathcal{V}_\alpha(f_1), \mathcal{S}(f_l, f_{l+1})$ and $\mathcal{U}(f_{n_y}, \beta)$, respectively. Thus $P_{(\alpha,\beta)}(n_y, a_y)$ is given by

$$\sum_{f_1=1}^{a_y-n_y+1} \sum_{f_2=f_1+1}^{a_y-n_y+2} \cdots \sum_{f_{n_y}=f_{n_y-1}+1}^{a_y} v_\alpha(f_1)s(f_1, f_2)\ldots s(f_{n_y-1}, n_y)u(f_{n_y}, \beta).$$

The computation of the probabilities $v_\alpha(f_1)$ and $s(f_l, f_{l+1})$ is similar to that in [8]. For their computation, as well as of $u(f_{n_y}, \beta)$ see [2].

Proposition 2. *The probabilities $v_\alpha(f_1), s(f_l, f_{l+1})$ and $u(f_{n_y}, \beta)$ are given as*

$$v_\alpha(f_1) = \begin{cases} 0 & f_1 \leq K - \alpha \\ \frac{\rho}{\rho+1}.\phi_{2f_1-K+\alpha-3}(\alpha+1, K) & o.w. \end{cases} \qquad \alpha \neq K, \tag{7}$$

$$v_K(f_1) = \begin{cases} 1 & f_1 = 1 \\ 0 & o.w. \end{cases}, \qquad s(f_l, f_{l+1}) = \frac{\rho}{\rho+1}.\phi_{2(f_{l+1}-f_l-1)}(K, K) \tag{8}$$

$$u(f_{n_y}, \beta) = \begin{cases} \phi_{2(a_y-f_{n_y})+K-\beta}(K, \beta) & f_{n_y} < a_y \\ 1 & f_{n_y} = a_y \text{ and } \beta = K \\ 0 & f_{n_y} = a_y \text{ and } \beta \neq K \end{cases} \tag{9}$$

where $\phi_\eta(\alpha,\beta)$ is defined as the probability of a path that starts with α packets in the buffer, ends with β packets in the buffer and consists of η events (arrivals and departures) and is defined as $\phi_\eta(\alpha,\beta) = \epsilon_\eta(\alpha,\beta) + \sum_{r=1}^{\mathcal{H}} W_\alpha Y^{r-1} Z^{\mathcal{T}}$, where, for $\alpha \geq 1, \beta \geq 1$ where $\epsilon_\eta(\alpha,\beta)$ is given by

$$\sum_{\Upsilon} \left[\binom{\eta}{\frac{\eta+\alpha-\beta}{2} - \Upsilon(K+1)} - \binom{\eta}{\frac{\eta-\alpha-\beta}{2} - \Upsilon(K+1)} \right] \left(\frac{\rho}{1+\rho}\right)^{\frac{\eta-\alpha+\beta}{2}} \left(\frac{1}{1+\rho}\right)^{\frac{\eta+\alpha-\beta}{2}},$$

$$\begin{aligned} W_\alpha &= \left(\epsilon_\alpha(\alpha,0), \epsilon_{\alpha+2}(\alpha,0), \ldots, \epsilon_{\alpha+2(\mathcal{H}-1)}(\alpha,0)\right) \\ Z &= \left(\epsilon_{\eta-\alpha}(0,\beta), \epsilon_{\eta-\alpha-2}(0,\beta), \ldots, \epsilon_{\eta-\alpha-2(\mathcal{H}-1)}(0,\beta)\right) \\ Y &= \begin{pmatrix} 0 & \epsilon_2(0,0) & \epsilon_4(0,0) & \ldots & \epsilon_{2(\mathcal{H}-1)}(0,0) \\ 0 & 0 & \epsilon_2(0,0) & \ldots & \epsilon_{2(\mathcal{H}-2)}(0,0) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & \epsilon_2(0,0) \\ 0 & 0 & 0 & \ldots & 0 \end{pmatrix}, \qquad \mathcal{H} = 1 + \frac{\eta-\alpha-\beta}{2} \end{aligned}$$

and $\epsilon_\eta(0,\beta) = \epsilon_{\eta-1}(1,\beta)$, $\beta \geq 1$, $\epsilon_\eta(\alpha,0) = \frac{1}{1+\rho}\epsilon_{\eta-1}(\alpha,1)$, $\alpha \geq 1$, $\epsilon_\eta(0,0) = \frac{1}{1+\rho}\epsilon_{\eta-2}(1,1)$, where $-\infty < \Upsilon < \infty$ takes on values in the sum in the definition of $\epsilon_\eta(\alpha,\beta)$ in (10) so that the binomial coefficients are proper, for e.g. in the first sum in (10) $\frac{\eta+\alpha-\beta}{2} > \Upsilon(K+1)$ and $\eta > \frac{\eta+\alpha-\beta}{2} - \Upsilon(K+1)$.

We also need the probability of the evolution of a path after the end of interval A_i and before the start of interval A_{i+1} and having $k_i(\geq 1)$ packets lost by random losses. Observe that the duration of this random loss event has the distribution of the sum of $k_i + 1$ independent $\exp(\lambda)$ distributed random variables, i.e., Erlang$(k_i + 1, \lambda)$. Let X_i be the number of service completions $\exp(\mu)$ in an interval with distribution $F * F * \ldots (k - times) = F^{*k}$ where $F \sim \exp(\lambda)$ and $*$ denotes the convolution operation. Then the probability that A_i ends with β_1 packets (including the last arrival in the interval A_i) in the buffer and A_{i+1} starts with β_2 packets (not including the first arrival in the interval A_{i+1}) in the buffer and has k_i random losses can be written as

$$P(X_i = \beta_1 - \beta_2, k_i) = \begin{cases} \int_0^\infty \frac{e^{-\mu s}(\mu s)^{(\beta_1-\beta_2)}}{(\beta_1-\beta_2)!} dF^{*(k_i+1)}(s) & \text{if } 0 < \beta_2 \leq \beta_1 \\ \sum_{m=\beta_1}^\infty \int_0^\infty \frac{e^{-\mu s}(\mu s)^m}{m!} dF^{*(k_i+1)}(s) & \text{if } \beta_2 = 0 \\ 0 & \beta_2 > \beta_1. \end{cases}$$

Remark 2. Indeed, the end of service times are a Poisson process with intensity μ. The PGF of the number of such points during a fix interval T is $G(z) = \exp(-\mu(1-z)T)$. If T is a random interval then it is $G(z) = E[\exp(-\mu(1-z)T] = T^*(\mu(1-z)$ where $T^*(s)$ is the Laplace Stieltjes transform of T. If T were exponential (λ) then this would give

$$G(z) = \frac{\lambda}{\lambda + \mu(1-z)} = \frac{1}{z}\frac{\theta z}{1-(1-p)z} \text{ where } \theta = \frac{\lambda}{\lambda+\mu} = \frac{\rho}{1-\rho}.$$

We see that $G(z)$ is the PGF of $Y = X - 1$ where X has a geometric distribution with parameter θ, so $P(Y = n) = (1-\theta)^n\theta$. The number of points in an Erlang($k_i + 1, \lambda$) RV, say X_i, has thus the distribution of the convolution of $k_i + 1$ copies of Y, which gives:

$$P(X_i = n) = \sum_{y_1+\ldots+y_n = k_i+1} \frac{(k_i+1)!}{y_1!y_2!\ldots y_n!}\theta^n(1-\theta)^{k_i+1}$$

This can now be used to for the expressions in (10).

We will now consider a path that starts with i packets in the buffer, in which out of n packets in a frame, j_r packets are lost by random losses j_c packets are lost by congestion losses, $j_c + j_r = j$ and has T *intervals.* Let r_i be the position of the ith random loss. Let $P_p^i(j_c, j_r, T, n)$ be the probability of such a path [3]. Then for $r_1 \neq 1$ and $r_{j_r} \neq n$,

$$\begin{aligned}
&P_p^i(j_c, j_r, T, n)\\
&= r^{j_r}(1-r)^{n-j_r} \sum_{\beta_g=0, 0\le g\le T-1}^{\beta_g=K} \; \sum_{\alpha_h=1, 0\le h\le T-1}^{\alpha_h=K} \; \sum_{r_1=2}^{n-j_r} \sum_{k_1=1}^{j_r} \sum_{k_2=1}^{j_r-k_1} \cdots \sum_{k_{T-2}=1}^{j_r-\sum_{h=1}^{T-3} k_h}\\
&\quad \sum_{a_2=1}^{n-j_r-a_1} \; \sum_{a_3=1}^{n-j_r-\sum_{i=1}^{2} a_i} \cdots \sum_{a_{T-1}=1}^{n-j_r-\sum_{i=1}^{T-2} a_i} \; \sum_{n_1=0}^{\min(r_1-2, j_c)} \; \sum_{n_2=0}^{\min(a_2, j_c-n_1)} \cdots \sum_{n_{T-1}=0}^{\min(a_{T-1}, j_c-\sum_{h=1}^{T-2} n_h)}\\
&\quad P_{(i,\beta_0)}(n_1, a_1) P(X_1 = \beta_0 - \alpha_1, k_1) P_{(\alpha_1,\beta_1)}(n_2, a_2)\\
&\quad P(X_2 = \beta_1 - \alpha_2, k_2) \ldots P_{(\alpha_{T-2},\beta_{T-2})}(n_{T-1}, a_{T-1}) P(X_{T-1} = \beta_{T-2} - \alpha_{T-1}, k_{T-1})\\
&\quad P_{(\alpha_{T-1},\beta_{T-1})}(n_T, a_T).
\end{aligned}$$

where $\sum_{k=1}^{i} f_k = 0$ for $i \le 0$ and $a_1 = r_1 - 1$, $a_T = n - j_r - \sum_{i=1}^{T-1} a_i$, $k_{T-1} = j_r - \sum_{h=1}^{T-2} k_h$, $n_T = j_c - \sum_{h=1}^{T-1} n_h$. One can similarly obtain expressions for the other cases ($j_r \le 2$) and/or $r_1 \neq 1$ and/or $r_{j_r} \neq n$ etc. Having obtained the expressions we have

$$P_p^i(j_c, j_r, n) = \sum_T P_p^i(j_c, j_r, T, n) \text{ and } P_p^i(j, n) = \binom{j}{j_c} P_p^i(j_c, j_r, n).$$

And finally, $P_p(j, n) = \sum_{i=0}^{K} \Pi(i) P_p^i(j, n)$. The probability $P_p(j, n)$ here is the same as the probability $P(j, n)$ in Sec. 3.

6 Conclusion

We have studied the steady state loss probabilities of messages in an $M/M/1/K$ queue where there are both random losses and congestion losses using an algebraic approach involving generating functions and a second approach based on ballot theorems. The explicit expressions we obtained allowed us to investigate numerically when it is profitable to add FEC, and what should the optimal block size be when we add a single redundant packet per block (e.g. using a XOR operation).

[3] We use the subscript p to distinguish the notation from Sec. 3

Appendix: Proof of Proposition 1

Define $\pi_{j,n}(x) \overset{\Delta}{=} \sum_{i=0}^{K} x^i P_i^a(j,n),\ n \geq 1,\ j \geq 0$. (3) implies for $n \geq 2$, that

$$\pi_{j,n}(x) = \bar{r} \sum_{i=0}^{K-1} x^i \sum_{k=0}^{i+1} Q_{i+1}(k) P_{i+1-k}^a(j, n-1)$$
$$+ r \sum_{i=0}^{K-1} x^i \sum_{k=0}^{i} Q_i(k) P_{i-k}^a(j-1, n-1) + x^K \sum_{k=0}^{K} Q_K(k) P_{K-k}^a(j-1, n-1).$$

We substitute (3) in the last equation, introduce $\pi_{j,n}(x)$ and also use the facts that $\pi_{j,n}(0) = P_0^a(j,n)$ and $1 - \rho\alpha = \alpha$. We then obtain for $n \geq 2$, $j \geq 1$, after some algebra [2]

$$\begin{aligned}\pi_{j,n}(x) &= \frac{\bar{r}\rho\alpha^2}{1-\alpha x}\left(\frac{1}{\alpha x}\pi_{j,n-1}(x) - (\alpha x)^K \pi_{j,n-1}(\alpha^{-1})\right)\\ &-\frac{\bar{r}\rho\alpha^2}{1-\alpha x}\left(\frac{1}{\alpha x} - (\alpha x)^K\right)\pi_{j,n-1}(0) + \bar{r}\alpha\frac{1-(\alpha x)^K}{1-\alpha x}\pi_{j,n-1}(0)\\ &+r\frac{\rho\alpha}{1-\alpha x}\left(\pi_{j-1,n-1}(x) - (\alpha x)^K \pi_{j-1,n-1}(\alpha^{-1})\right) \qquad (10)\\ &+r\alpha\frac{1-(\alpha x)^K}{1-\alpha x}\pi_{j-1,n-1}(0) + \alpha\rho(\alpha x)^K \pi_{j-1,n-1}(\alpha^{-1}) + \alpha(\alpha x)^K \pi_{j-1,n-1}(0).\end{aligned}$$

Define, with some abuse of notation, the generating function of $P_i^a(j,n)$ $\pi(x,y,z) \overset{\Delta}{=} \sum_{j=0}^{\infty}\sum_{n=1}^{\infty} y^j z^{n-1} \pi_{j,n}(x)$. When we fix y and $|z| < 1$, the above generating function is polynomial in x, and therefore an analytic function. In order to use (10), which holds only for $n \geq 2$ and $j \geq 1$, we note that $\sum_{j=1}^{\infty}\sum_{n=2}^{\infty} y^j z^{n-1}\pi_{j,n}(x) = \pi(x,y,z) - \pi(x,0,z) - \pi(x,y,0) + \pi(x,0,0)$. We obtain after some algebra [2]

$$\begin{aligned}&\pi(x,y,z) - \pi(x,0,z)\\ &= yx^K + r\frac{1-x^K}{1-x}y + \bar{r}\frac{\rho\alpha^2 z}{(1-\alpha x)\alpha x}\left[\pi(x,y,z) - \pi(x,0,z)\right] + \frac{r\rho\alpha y z}{1-\alpha x}\pi(x,y,z)\\ &+ \rho\alpha(\alpha x)^K\left(y - \frac{(\bar{r}\alpha + ry)}{1-\alpha x}\right) z\left[\pi(\alpha^{-1},y,z) + \pi(0,y,z)/\rho\right]\\ &+ \frac{\bar{r}\alpha^2(x-\rho)}{(1-\alpha x)\alpha x} z\left[\pi(0,y,z) - \pi(0,0,z)\right]\\ &+ \frac{\bar{r}\rho\alpha^2(\alpha x)^K}{1-\alpha x} z\left[\pi(\alpha^{-1},0,z) + \pi(0,0,z)/\rho\right] + \frac{r\alpha y z}{1-\alpha x}(\alpha x)^K \pi(0,y,z). \qquad (11)\end{aligned}$$

We note that in order to establish the proof of Proposition 1, it follows from (1) that it suffices to obtain $\pi(x,y,z)$ at $x = \rho$, since $q(y,z) = R_K\pi(\rho,y,z)$. From (11), we have

$$\left[\pi(\rho,y,z) - \pi(\rho,0,z)\right](1 - (\bar{r} + r\rho y)z) = y\rho^K + r\frac{1-\rho^K}{1-\rho}y$$

$$+z\left(y-\bar{r}-\frac{ry}{\alpha}\right)(\rho\alpha)^{K+1}\left[\pi(\alpha^{-1},y,z)+\pi(0,y,z)/\rho\right]$$
$$+z\bar{r}(\rho\alpha)^{K+1}\left[\pi(\alpha^{-1},0,z)+\pi(0,0,z)/\rho\right]+r\rho yz\left[\pi(\rho,0,z)+\frac{(\alpha\rho)^K}{\rho}\pi(0,y,z)\right].$$

To compute the function $\pi(\rho,y,z)$ it suffices to compute the functions in the square brackets as well as $\pi(\rho,0,z)$. To do that, we first compute $\pi_{0,n}$ by proceeding in the same manner as in (10). Since $P_K^a(0,n)=0$ we have for $n\geq 2$,

$$\pi_{0,n}(x)=\bar{r}\tfrac{\rho\alpha^2}{1-\alpha x}\tfrac{1}{\alpha x}\pi_{0,n-1}(x)-\bar{r}\tfrac{\rho\alpha^2}{1-\alpha x}(\alpha x)^K\pi_{0,n-1}(\alpha^{-1})$$
$$+\bar{r}\alpha\tfrac{1-(\alpha x)^K}{1-\alpha x}\pi_{0,n-1}(0)-\bar{r}\tfrac{\rho\alpha^2}{1-\alpha x}\left(\tfrac{1}{\alpha x}-(\alpha x)^K\right)\pi_{0,n-1}(0).$$

Taking the generating function of both sides and substituting $\pi(x,0,0)=\bar{r}\frac{1-x^K}{1-x}$, we get

$$(1-\alpha x)\alpha x\pi(x,0,z)=\bar{r}\frac{1-x^K}{1-x}(1-\alpha x)\alpha x+\bar{r}\rho\alpha^2 z\pi(x,0,z)$$
$$-\bar{r}\rho\alpha^2(\alpha x)^{K+1}z\left[\pi(\alpha^{-1},0,z)+\pi(0,0,z)/\rho\right]+\bar{r}\alpha^2(x-\rho)z\pi(0,0,z). \quad (12)$$

From (11), we have

$$\left((1-\alpha x)\alpha x-\rho\alpha^2\bar{r}z\right)\left[\pi(x,y,z)-\pi(x,0,z)\right]$$
$$=(1-\alpha x)\alpha y x^{K+1}+(1-\alpha x)\alpha x r\frac{1-x^K}{1-x}y$$
$$+z\rho\alpha(\alpha x)^{K+1}\left[(y(1-\alpha x)-(\bar{r}\alpha+ry)\right]\times\left[\pi(\alpha^{-1},y,z)+\pi(0,y,z)/\rho\right]$$
$$+\bar{r}\rho\alpha^2(\alpha x)^{K+1}z\left[\pi(\alpha^{-1},0,z)+\pi(0,0,z)/\rho\right]+\alpha^2 r\rho xyz\pi(x,y,z)$$
$$+\alpha^2\bar{r}(x-\rho)z\left[\pi(0,y,z)-\pi(0,0,z)\right]+\alpha^2 rxyz(\alpha x)^K\pi(0,y,z). \quad (13)$$

Substituting (12) in (13) yields

$$\left((1-\alpha x)\alpha x-\rho\alpha^2\left(\bar{r}z+rxyz\right)\right)\pi(x,y,z)$$
$$=\ (1-\alpha x)\alpha y x^{K+1}+(1-\alpha x)\alpha x\left(ry+\bar{r}\right)\frac{1-x^K}{1-x}$$
$$+z\rho\alpha(\alpha x)^{K+1}\left(y\left(\bar{r}-\alpha x\right)-\bar{r}\alpha\right)\times\left[\pi(\alpha^{-1},y,z)+\pi(0,y,z)/\rho\right]$$
$$+z\alpha^2\left(\bar{r}(x-\rho)+rxy(\alpha x)^K\right)\pi(0,y,z). \quad (14)$$

For each $i=1,2$, when $x=x_i(y,z)$, the term that multiplies $\pi(x,y,z)$ in the left hand side of equation (14) vanishes. Since $\pi(x,y,z)$ is polynomial in x and therefore analytic in x, the left hand side of (14) vanishes at $x=x_i(y,z)$. Thus by substituting x_i for x into (14), we obtain two equations (4) with two unknowns: $A(y,z)=\left[\pi(\alpha^{-1},y,z)+\pi(0,y,z)/\rho\right]$ and $B(y,z)=\pi(0,y,z)$. Equation (3) of the proposition, finally, follows from (14) with $x=\rho$ and since $q(y,z)=R_K\pi(\rho,y,z)$.

References

1. E. Altman, C. Barakat and V. M. Ramos Ramos, "Queuing analysis of simple FEC schemes for Voice over IP", to appear at *Communications Networks*, 2002.
2. P. Dube, O. Ait-Hellal and E. Altman, "On Loss Probabilities in Presence of Redundant Packets with Random Drop", INRIA Research Report, 2002.
3. O. Ait-Hellal, E. Altman, A. Jean-Marie, I. A. Kurkova, "On Loss Probabilities in Presence of Redundant Packets and Several Traffic Sources", *Perf. Eval.*, **36-37**, pp. 485-518, 1999.
4. I. Cidon, A. Khamisy and M. Sidi, "Analysis of Packet Loss Process in High-Speed Networks", *IEEE Trans. IT*, Vol. 39, No. 1, pp. 98-108, 1993.
5. L. Takacs, "Combinatorial Methods in the Theory of Stochastic Processes", New York,Wiley,1967.
6. N. Shacham and P. McKenney, "Packet Recovery in High-Speed Networks Using Coding and Buffer Management", *INFOCOM '90*, San Francisco, CA, pp. 124-131.
7. I. Cidon, R. Guerin, I. Kessler and A. Khamisy, "Analysis of a Statistical Multiplexer with Generalized Periodic Sources", *Queueing Systems*, **20**, pp. 139-169, 1995.
8. O. Gurewitz, M. Sidi and I. Cidon, "The Ballot Theorem Strikes Again: Packet Loss Process Distribution", *IEEE Trans. on Information Theory*, **46**, Nov 2000.

Performance Analysis of a GI-G-1 Preemptive Resume Priority Buffer

Joris Walraevens, Bart Steyaert, and Herwig Bruneel

SMACS Research Group
Ghent University, Vakgroep TELIN (TW07V)
Sint-Pietersnieuwstraat 41, B-9000 Gent, Belgium.
Phone: 0032-9-2648902
Fax: 0032-9-2644295
{jw,bs,hb}@telin.rug.ac.be

Abstract. In this paper, we have analyzed a discrete-time $GI-G-1$ queue with a preemptive resume priority scheduling and two priority classes. We have derived the joint generating function of the system contents of both classes and the generating functions of the delay of both classes. These pgf's are not explicitly found, but we have proven that the moments of the distributions can be found explicitly in terms of the system parameters. We have shown the impact of priority scheduling on the performance characteristics by some numerical examples.

1 Introduction

In recent years, there has been much interest devoted to incorporating multimedia applications in packet networks (e.g., IP networks). Different types of traffic need different QoS standards, but share the same network resources, such as buffers and bandwidth. For real-time applications, it is important that mean delay and delay-jitter are bounded, while for non real-time applications, the Loss Ratio (LR) is the restrictive quantity.

In general, one can distinguish two priority strategies, which will be referred to as Delay priority and Loss priority. Delay priority schemes attempt to guarantee acceptable delay boundaries to delay-sensitive traffic (such as voice/video). This can for instance be achieved by giving it HOL priority over non-delay-sensitive traffic. Several types of Delay priority (or scheduling) schemes have been proposed and analyzed, each with their own specific algorithmic and computational complexity (see e.g. [5, 8] and the references therein). On the other hand, Loss priority schemes attempt to minimize the packet loss of loss-sensitive traffic (such as data). An overview and classification of some Loss priority (or discarding) strategies can be found in [5, 3].

In this paper, we will focus on the effect of a specific Delay priority scheme, i.e., the preemptive resume priority scheduling discipline. We assume that delay-sensitive traffic has preemptive priority over delay-insensitive traffic, i.e., when the server becomes empty, a packet of delay-sensitive traffic, when available, will always be scheduled next. In the remaining, we will refer to the delay-sensitive

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 745–756, 2002.

and delay-insensitive traffic as high and low priority traffic respectively. Newly arriving high priority traffic interrupts transmission of a low priority packet that has already commenced, and the interrupted low priority packet can resume its transmission when all the high priority traffic has left the system.

In the literature, there have been a number of contributions with respect to HOL priority scheduling. An overview of some basic HOL priority queueing models can be found in Kleinrock [4], Miller [7] and Takagi [9] and the references therein. Preemptive resume priority queues have been analyzed in Machihara [6], Takine et al. [10] and Walraevens et al. [11]. Machihara [6] analyzes waiting times when high priority arrivals are distributed according to a MAP process. Takine [10] studies the waiting times of customers arriving to a queue according to independent MAP processes. Finally, Walraevens [11] analyzes system contents and packet delay when the length of high priority packets are generally distributed and the length of low priority packets are geometrically distributed.

In this paper, we analyze the system contents and packet delay of high priority and low priority traffic in a discrete-time single-server buffer for a preemptive resume priority scheme and per-slot i.i.d. arrivals. The transmission times of the packets are assumed to be generally distributed. These distribution can be class-dependent, i.e., the transmission times of the high priority packets can be different from those of the low priority packets. We will demonstrate that an analysis based on generating functions is extremely suitable for modelling this type of buffers with a priority scheduling discipline. From these generating functions, expressions for some interesting performance measures (such as moments of system contents and packet delay of both classes) can be calculated.

The remainder of this paper is structured as follows. In the following section, we present the mathematical model. In sections 3 and 4, we will then analyze the steady-state system contents and packet delay of both classes. In section 5, we give expressions for some moments of the system contents and packet delay of both classes. Some numerical examples are treated in section 6. Finally, some conclusions are formulated in section 7.

2 Mathematical Model

We consider a discrete-time single-server system with infinite buffer space. Time is assumed to be slotted. There are two types of packets arriving to the system, namely packets of class 1 and packets of class 2. The number of arrivals of class j during slot k are i.i.d. and are denoted by $a_{j,k}$ $(j = 1, 2)$. Their joint probability mass distribution is defined as $a(m, n) \triangleq \text{Prob}[a_{1,k} = m, a_{2,k} = n]$. Note that the number of arrivals of both classes can be correlated during one slot. The joint probability generating function (pgf) of $a_{1,k}$ and $a_{2,k}$ is defined as $A(z_1, z_2) \triangleq E[z_1^{a_{1,k}} z_2^{a_{2,k}}] = \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} a(m, n) z_1^m z_2^n$. The marginal pgf's of the number of arrivals of class j are denoted by $A_j(z)$ $(j = 1, 2)$ and are given by $A(z, 1)$ and $A(1, z)$ respectively. We will furthermore denote the mean arrival rate of class j packets during a slot by $\lambda_j \triangleq E[a_{j,k}] = A'_j(1)$ $(j = 1, 2)$.

The service times of the class j packets, i.e., the number of slots a class j packet is effectively being served, are i.i.d. and generally distributed and their pgf is denoted by $S_j(z)$ $(j = 1, 2)$. The mean service time of a class j packet is given by μ_j $(j = 1, 2)$.

The class 1 packets are assumed to have preemptive resume priority over the class 2 packets and within one class the scheduling is FCFS. The load offered by class j packets is given by $\rho_j \triangleq \lambda_j\mu_j$. The total load is then given by $\rho_T \triangleq \rho_1+\rho_2$. We assume a stable system, i.e., $\rho_T < 1$.

3 System Contents

We denote the system contents of class j packets at the beginning of slot k by $u_{j,k}$ $(j = 1, 2)$. Their joint pgf is defined as $U_k(z_1, z_2) \triangleq E\left[z_1^{u_{1,k}} z_2^{u_{2,k}}\right]$. Since service times of both classes are generally distributed, the set $\{u_{1,k}, u_{2,k}\}$ does not form a Markov chain. Therefore, we introduce two new stochastic variables $r_{j,k}$ $(j = 1, 2)$ as follows: $r_{1,k}$ indicates the remaining number of slots needed to transmit the class 1 packet in service at the beginning of slot k, if $u_{1,k} > 0$, and $r_{1,k} = 0$ if $u_{1,k} = 0$; $r_{2,k}$ indicates the remaining number of slots service time of the class 2 packet longest in the system at the beginning of slot k, if $u_{2,k} > 0$, and $r_{2,k} = 0$ if $u_{2,k} = 0$. With these definitions, $\{r_{1,k}, u_{1,k}, r_{2,k}, u_{2,k}\}$ is easily seen to constitute a Markovian state description of the system at the beginning of slot k. If $s^*_{j,k}$ $(j = 1, 2)$ indicates the service time of the next class j packet to receive service at the beginning of slot k, the following system equations can be established:

1. If $r_{1,k} = 0$ (and hence $u_{1,k} = 0$):
   a) If $r_{2,k} = 0$ (and hence $u_{2,k} = 0$):

$$u_{j,k+1} = a_{j,k}\ ;\ r_{j,k+1} = \begin{cases} 0 & \text{if } a_{j,k} = 0 \\ s^*_{j,k} & \text{if } a_{j,k} > 0 \end{cases},$$

   with $j = 1, 2$. The only packets present in the system at the beginning of slot $k + 1$ are the packets that arrive during the previous slot. If there have been new arrivals of class j packets during slot k, the remaining number of slots needed to service the first class j packet is that packet's full service time.

   b) If $r_{2,k} = 1$:

$$\begin{aligned} u_{1,k+1} &= a_{1,k} &;\ u_{2,k+1} &= u_{2,k} - 1 + a_{2,k}; \\ r_{1,k+1} &= \begin{cases} 0 & \text{if } a_{1,k} = 0 \\ s^*_{1,k} & \text{if } a_{1,k} > 0 \end{cases} &;\ r_{2,k+1} &= \begin{cases} 0 & \text{if } u_{2,k} - 1 + a_{2,k} = 0 \\ s^*_{2,k} & \text{if } u_{2,k} - 1 + a_{2,k} > 0 \end{cases}, \end{aligned}$$

   i.e., the class 2 packet in service at the beginning of slot k leaves the system at the end of slot k.

c) If $r_{2,k} > 1$:

$$\begin{aligned} u_{1,k+1} &= a_{1,k} &;\ & u_{2,k+1} = u_{2,k} + a_{2,k}; \\ r_{1,k+1} &= \begin{cases} 0 & \text{if } a_{1,k} = 0 \\ s^*_{1,k} & \text{if } a_{1,k} > 0 \end{cases} &;\ & r_{2,k+1} = r_{2,k} - 1, \end{aligned}$$

i.e., the class 2 packet in service at the beginning of slot k remains in the system (not necessarily in the server - because of the preemptive priority scheduling discipline, it can only remain in the server if there are no new class 1 arrivals). Its remaining service time is decreased by one.

2. If $r_{1,k} = 1$:

a) If $r_{2,k} = 0$ (and hence $u_{2,k} = 0$):

$$\begin{aligned} u_{1,k+1} &= u_{1,k} - 1 + a_{1,k} &;\ & u_{2,k+1} = a_{2,k}; \\ r_{1,k+1} &= \begin{cases} 0 & \text{if } u_{1,k} - 1 + a_{1,k} = 0 \\ s^*_{1,k} & \text{if } u_{1,k} - 1 + a_{1,k} > 0 \end{cases} &;\ & r_{2,k+1} = \begin{cases} 0 & \text{if } a_{2,k} = 0 \\ s^*_{2,k} & \text{if } a_{2,k} > 0 \end{cases}, \end{aligned}$$

i.e., the class 1 packet in service at the beginning of slot k, leaves the system at the end of slot k. There were no class 2 packets in the system at the beginning of slot k.

b) If $r_{2,k} > 0$:

$$\begin{aligned} u_{1,k+1} &= u_{1,k} - 1 + a_{1,k} &;\ & u_{2,k+1} = u_{2,k} + a_{2,k}; \\ r_{1,k+1} &= \begin{cases} 0 & \text{if } u_{1,k} - 1 + a_{1,k} = 0 \\ s^*_{1,k} & \text{if } u_{1,k} - 1 + a_{1,k} > 0 \end{cases} &;\ & r_{2,k+1} = r_{2,k}, \end{aligned}$$

i.e., the class 1 packet in service at the beginning of slot k, leaves the system at the end of slot k. The remaining service of the class 2 packet longest in the system stays the same.

3. If $r_{1,k} > 1$:

a) If $r_{2,k} = 0$ (and hence $u_{2,k} = 0$):

$$\begin{aligned} u_{1,k+1} &= u_{1,k} + a_{1,k} &;\ & u_{2,k+1} = a_{2,k}; \\ r_{1,k+1} &= r_{1,k} - 1 &;\ & r_{2,k+1} = \begin{cases} 0 & \text{if } a_{2,k} = 0 \\ s^*_{2,k} & \text{if } a_{2,k} > 0 \end{cases}, \end{aligned}$$

i.e., the class 1 packet in service at the beginning of slot k stays in the server at the beginning of slot $k+1$. Its remaining service is decreased by one.

b) If $r_{2,k} > 0$:

$$\begin{aligned} u_{j,k+1} &= u_{j,k} + a_{j,k}; \\ r_{1,k+1} &= r_{1,k} - 1 \quad ; \quad r_{2,k+1} = r_{2,k}, \end{aligned}$$

with $j = 1, 2$. The difference with the previous case is that there was at least one class 2 packet in the system at the beginning of slot k.

We define $\mathrm{E}[X\{Y\}]$ as $\mathrm{E}[X|Y]\mathrm{Prob}[Y]$ in the remainder. We furthermore define $P_k(x_1, z_1, x_2, z_2)$ as the joint pgf of the state vector $(r_{1,k}, u_{1,k}, r_{2,k}, u_{2,k})$, i.e., $P_k(x_1, z_1, x_2, z_2) \triangleq E[x_1^{r_{1,k}} z_1^{u_{1,k}} x_2^{r_{2,k}} z_2^{u_{2,k}}]$. We assume that the system is stable (implying that the equilibrium condition requires that $\rho_T < 1$) and as a result $P_k(x_1, z_1, x_2, z_2)$ and $P_{k+1}(x_1, z_1, x_2, z_2)$ converge both to a common steady state value $P(x_1, z_1, x_2, z_2) = \lim_{k\to\infty} P_k(x_1, z_1, x_2, z_2)$. Using the system equations, we can constitute a relation between $P_k(x_1, z_1, x_2, z_2)$ and $P_{k+1}(x_1, z_1, x_2, z_2)$. By taking the $k \to \infty$ limit in this relation between $P_k(x_1, z_1, x_2, z_2)$ and $P_{k+1}(x_1, z_1, x_2, z_2)$ we obtain:

$$\begin{aligned}
&[x_1 - A(z_1, z_2)]P(x_1, z_1, x_2, z_2)\\
&= \Bigg[x_1 A(0, 0)(1 - S_1(x_1))(1 - S_2(x_2)) + \frac{x_1}{x_2} A(0, z_2)(1 - S_1(x_1))(x_2 S_2(x_2) - 1)\\
&\quad + A(z_1, 0)(x_1 S_1(x_1) - 1)(1 - S_2(x_2))\\
&\quad + \frac{1}{x_2} A(z_1, z_2)(x_1 x_2 S_1(x_1) S_2(x_2) - x_1 S_1(x_1) - x_2 S_2(x_2) + x_2)\Bigg] P(0, 0, 0, 0)\\
&\quad + x_1[A(0, 0)(1 - S_1(x_1)) + A(z_1, 0)S_1(x_1)](1 - S_2(x_2))R_2(0)\\
&\quad + x_1(A(0, z_2) - A(0, 0))(1 - S_1(x_1))(S_2(x_2) - 1)R_1(0, 0, 0)\\
&\quad + (A(z_1, z_2) - A(z_1, 0))(S_2(x_2) - 1)P(x_1, z_1, 0, 0)\\
&\quad + x_1(A(z_1, z_2) - A(z_1, 0))(z_1 - S_1(x_1))(1 - S_2(x_2))R_1(z_1, 0, 0)\\
&\quad + \frac{1}{x_2}[x_1 A(0, z_2)(1 - S_1(x_1)) + A(z_1, z_2)(x_1 S_1(x_1) - x_2)]P(0, 0, x_2, z_2)\\
&\quad + x_1[A(0, z_2)(1 - S_1(x_1)) + A(z_1, z_2)S_1(x_1)](S_2(x_2) - z_2)R_2(z_2)\\
&\quad + x_1 A(0, z_2)(1 - S_1(x_1))R_1(0, x_2, z_2) + x_1 A(z_1, z_2)(S_1(x_1) - z_1)R_1(z_1, x_2, z_2)
\end{aligned}$$

with functions $R_1(z_1, x_2, z_2) \triangleq \lim_{k\to\infty} \mathrm{E}\left[z_1^{u_{1,k}-1} x_2^{r_{2,k}} z_2^{u_{2,k}}\{r_{1,k} = 1\}\right]$ and $R_2(z_2) \triangleq \lim_{k\to\infty} \mathrm{E}\left[z_2^{u_{2,k}-1}\{r_{1,k} = u_{1,k} = 0, r_{2,k} = 1\}\right]$. It now remains for us to determine the functions $P(x_1, z_1, 0, 0)$, $P(0, 0, x_2, z_2)$, $R_2(z_2)$, $R_1(z_1, x_2, z_2)$ and the unknown parameters $P(0, 0, 0, 0)$, $R_2(0)$ and $R_1(0, 0, 0)$. Using generating functions techniques (a.o. Rouché's theorem), we can ultimately calculate a fully determined version for $P(x_1, z_1, x_2, z_2)$ (calculations are omitted due to page limitations):

$$\begin{aligned}
&P(x_1, z_1, x_2, z_2) = (1 - \rho_T)\\
&\Bigg[1 + \frac{x_1 z_1 (A(z_1, 0) - A(Y(0), 0))(S_1(x_1) - S_1(A(z_1, 0)))(1 - S_2(x_2))}{A(Y(0), 0)(x_1 - A(z_1, 0))(z_1 - S_1(A(z_1, 0)))(z_1 - S_1(A(z_1, z_2)))}\\
&+ x_1 z_1 \frac{(A(z_1, z_2) - A(Y(z_2), z_2))(S_1(x_1) - S_1(A(z_1, z_2)))}{(x_1 - A(z_1, z_2))(z_1 - S_1(A(z_1, z_2)))(z_2 - S_2(A(Y(z_2), z_2)))}\\
&\left\{\frac{S_2(A(Y(z_2), z_2))(z_2 - S_2(x_2))}{A(Y(z_2), z_2)} - z_2 \frac{(1 - x_2)(S_2(x_2) - S_2(A(Y(z_2), z_2)))}{x_2 - A(Y(z_2) z_2)}\right\}\\
&- x_2 z_2 \frac{(1 - A(Y(z_2), z_2))(S_2(x_2) - S_2(A(Y(z_2), z_2)))}{(x_2 - A(Y(z_2), z_2))(z_2 - S_2(A(Y(z_2), z_2)))}\Bigg], \qquad (1)
\end{aligned}$$

with $Y(z)$ implicitly defined as $Y(z) \triangleq S_1(A(Y(z), z))$. From this pgf, several joint and marginal pgf's can be calculated. We can for instance calculate the joint pgf of the system contents of class j packets and the remaining service of the class j packet with the longest waiting time at the beginning of an arbitrary slot in steady-state defined as follows $P_j(x, z) \triangleq \lim\limits_{k\to\infty} \mathrm{E}\left[x^{r_{j,k}} z^{u_{j,k}}\right]$, $j = 1, 2$. $P_1(x_1, z_1)$ ($P_2(x_2, z_2)$ respectively) can then be found from equation (1) by substituting x_2 and z_2 (x_1 and z_1 respectively) by 1. More importantly, we can calculate the joint pgf of the steady-state system contents of class 1 and class 2 packets from equation (1). It is given by:

$$
\begin{aligned}
U(z_1, z_2) &\triangleq \lim_{k\to\infty} \mathrm{E}\left[z_1^{u_{1,k}} z_2^{u_{2,k}}\right] = P(1, z_1, 1, z_2) \\
&= (1 - \rho_T) \frac{S_2(A(Y(z_2), z_2))(z_2 - 1)}{z_2 - S_2(A(Y(z_2), z_2))} \\
&\qquad \left[1 + z_1 \frac{(A(z_1, z_2) - A(Y(z_2), z_2))(S_1(A(z_1, z_2)) - 1)}{A(Y(z_2), z_2)(A(z_1, z_2) - 1)(z_1 - S_1(A(z_1, z_2)))}\right] .
\end{aligned}
\tag{2}
$$

From the two-dimensional pgf $U(z_1, z_2)$, we can easily derive expressions for the pgf of the system contents of class 1 packets and class 2 packets at the beginning of an arbitrary slot from expression (2), yielding

$$
\begin{aligned}
U_1(z) &\triangleq \lim_{k\to\infty} \mathrm{E}\left[z^{u_{1,k}}\right] = U(z, 1) \\
&= (1 - \rho_1) \frac{S_1(A_1(z))(z - 1)}{z - S_1(A_1(z))} ;
\end{aligned}
\tag{3}
$$

$$
\begin{aligned}
U_2(z) &\triangleq \lim_{k\to\infty} \mathrm{E}\left[z^{u_{2,k}}\right] = U(1, z) \\
&= (1 - \rho_T) \frac{A_2(z)}{A(Y(z), z)} \frac{S_2(A(Y(z), z)(z - 1)}{z - S_2(A(Y(z), z))} \frac{1 - A(Y(z), z)}{1 - A_2(z)} .
\end{aligned}
\tag{4}
$$

4 Packet Delay

The packet delay is defined as the total amount of time a packet spends in the system, more precisely, the number of slots between the end of the packet's arrival slot and the end of its departure slot. We can analyze the packet delay of class 1 packets as if they are the only packets in the system. This is e.g. done in [1] and the pgf of the packet delay of class 1 packets is given by

$$
D_1(z) = \frac{1 - \rho_1}{\lambda_1} \frac{S_1(z)(z - 1)}{z - A_1(S_1(z))} \frac{1 - A_1(S_1(z))}{1 - S_1(z)} .
\tag{5}
$$

Because of the priority discipline, the analysis of the delay of the low priority class will be a bit more involved. We tag a class 2 packet that enters the buffer during slot k. Let us refer to the packets in the system at the end of slot k, but that have to be served before the tagged packet as the "primary packets". So, basically, the tagged class 2 packet can enter the server, when all primary packets and all class 1 packets that arrived after slot k are transmitted. In order to analyse the delay of the tagged class 2 packet, the number of class 1 packets and class 2 packets that are served between the arrival slot of the tagged class 2 packet and its departure slot is important, not the precise order in which they are served. Therefore, in order to facilitate the analysis, we will consider an equivalent virtual system with an altered service discipline. We assume that from slot k on, the order of service for class 1 packets (those in the queue at the end of slot k and newly arriving ones) is LCFS instead of FCFS in the equivalent system (the transmission of class 2 packets remains FCFS). So, a primary packet can enter the server, when the system becomes free (for the first time) of class 1 packets that arrived during and after the service time of the primary packet that predecessed it according to the new service discipline. Let $v_{1,m}^{(i)}$ denote the length of the time period during which the server is occupied by the m-th class 1 packet that arrives during slot i and its class 1 "successors", i.e., the time period starting at the beginning of the service of that packet and terminating when the system becomes free (for the first time) of class 1 packets which arrived during and after its service time. Analogously, let $v_{2,m}^{(i)}$ denote the length of the time period during which the server is occupied by the m-th class 2 packet that arrives during slot i and its class 1 "successors". The $v_{j,m}^{(i)}$'s ($j = 1, 2$) are called sub-busy periods, caused by the m-th class j packet that arrived during slot i. The service time of the tagged class 2 packet is denoted by s_2^*.

When the tagged class 2 packet arrives, the system is in one of the following states:

1. $r_{1,k} = 0$ (and hence $u_{1,k} = 0$):
   a) $r_{2,k} = 0$ (and hence $u_{2,k} = 0$):

$$d_2 = \sum_{j=1}^{2} \sum_{m=1}^{f_{j,k}} v_{j,m}^{(k)} + s_2^* + \sum_{i=1}^{s_2^*-1} \sum_{m=1}^{a_{1,l_i}} v_{j,m}^{(l_i)}, \tag{6}$$

   with $f_{j,k}$ defined as the number of class j packets arriving during slot k, but that have to be served before the tagged packet. Slots l_i are defined as the slots during which the tagged packet receives service ($i = 1, .., s_2^*$). $f_{1,k}$ class 1 primary packets and $f_{2,k}$ class 2 primary packets that arrived during slot k and their class 1 successors have to be served before the tagged class 2 packet. During the service time of the tagged class 2 packet, new class 1 packets may arrive, which interrupt the tagged packet's service. The last two terms take this part of the delay into account.

   b) $r_{2,k} > 0$:

$$d_2 = (r_{2,k} - 1) + \sum_{i=1}^{r_{2,k}-1} \sum_{m=1}^{a_{1,n_i}} v_{1,m}^{(n_i)} + \sum_{j=1}^{2} \sum_{m=1}^{f_{j,k}} v_{j,m}^{(k)} + \sum_{m=1}^{u_{2,k}-1} \tilde{v}_{2,m} \tag{7}$$
$$+ s_2^* + \sum_{i=1}^{s_2^*-1} \sum_{m=1}^{a_{1,l_i}} v_{j,m}^{(l_i)},$$

with the n_i-th slots ($i = 1, .., r_{2,k}-1$) the slots (after slot k) that the class 2 packet longest in the server receives service and the $\tilde{v}_{2,m}$'s are defined as the sub-busy periods, caused by the m-th class 2 packet already in the queue at the beginning of start slot l. The residual service time of the packet in service during slot k contributes in the first term, the sub-busy periods of the class 1 packets arriving during the residual service time contribute in the second term, the sub-busy periods of the class 1 and class 2 packets arriving during slot k, but that have to be served before the tagged class 2 packet contribute in the third term, the sub-busy periods of the class 2 packets already in the queue at the beginning of slot k contribute in the fourth term and finally the service time of the tagged class 2 packet itself and the sub-busy periods of the class 1 packets arriving during this service time (except for its last slot) contribute in the last two terms.

2. $r_{1,k} > 0$:
   a) $r_{2,k} = 0$ (and hence $u_{2,k} = 0$):

$$d_2 = (r_{1,k} - 1) + \sum_{i=1}^{r_{1,k}-1} \sum_{m=1}^{a_{1,k+i}} v_{1,m}^{(k+i)} + \sum_{j=1}^{2} \sum_{m=1}^{f_{j,k}} v_{j,m}^{(k)} + \sum_{m=1}^{u_{1,k}-1} \tilde{v}_{1,m} \quad (8)$$
$$+ s_2^* + \sum_{i=1}^{s_2^*-1} \sum_{m=1}^{a_{1,l_i}} v_{j,m}^{(l_i)},$$

with the $\tilde{v}_{1,m}$'s the sub-busy periods, caused by the m-th class 1 packet already in the queue at the beginning of slot k. The expression is almost the same as in the previous case, with the difference that a class 1 packet was being served during slot k.

   b) $r_{2,k} > 0$:

$$d_2 = (r_{1,k} - 1) + \sum_{i=1}^{r_{1,k}-1} \sum_{m=1}^{a_{1,k+i}} v_{1,m}^{(k+i)} + \sum_{j=1}^{2} \sum_{m=1}^{f_{j,k}} v_{j,m}^{(k)} + \sum_{m=1}^{u_{1,k}-1} \tilde{v}_{1,m} \quad (9)$$
$$+ r_{2,k} + \sum_{i=1}^{r_{2,k}} \sum_{m=1}^{a_{1,n_i}} v_{1,m}^{(n_i)} + \sum_{m=1}^{u_{2,l}-1} \tilde{v}_{2,m} + s_2^* + \sum_{i=1}^{s_2^*} \sum_{m=1}^{a_{1,l_i}} v_{j,m}^{(l_i)}.$$

This case is a combination of the former two cases.

Due to the initial assumptions and since the length of different sub-busy periods only depends on the number of class 1 packet arrivals during different slots and the service times of the corresponding primary packets, the sub-busy periods associated with the primary packets of class 1 and class 2 form a set of i.i.d. random variables and their pgf will be presented by $V_1(z)$ and $V_2(z)$ respectively. Notice that $f_{1,k}$ and $f_{2,k}$ are correlated; in section 2 it was explained that $a_{1,k}$ and $a_{2,k}$ may be correlated as well. Once again, applying a z-transform technique to equations (6)-(9) and taking into account the previous remarks, we can ultimately derive an expression for $D_2(z)$:

$$D_2(z) = \frac{1-\rho_T}{\lambda_2} \frac{S_2(z)(A(V_1(z), V_2(z)) - A_1(V_1(z)))}{zA_1(V_1(z)) - A(V_1(z), V_2(z))} \frac{1 - zA_1(V_1(z))}{1 - V_2(z)}. \tag{10}$$

Finally, we have to find expressions for $V_1(z)$ and $V_2(z)$. These pgf's satisfy the following relations:

$$V_j(z) = S_j(zA_1(V_1(z))), \tag{11}$$

with $j = 1, 2$. This can be understood as follows: when the m-th class j packet that arrived during slot i enters service, $v_{j,m}^{(i)}$ consists of two parts: the service time of that packet itself, and the service times of the class 1 packets that arrive during its service time and of their class 1 successors. This leads to equation (11).

5 Calculation of Moments

The functions $Y(z)$, $V_1(z)$ and $V_2(z)$ can only be explicitly found in case of some simple arrival processes. Their derivatives for $z = 1$, necessary to calculate the moments of the system contents and the packet delay, on the contrary, can be calculated in closed-form. Let us define λ_{ij} and μ_{jj} as $\lambda_{ij} \triangleq \left.\frac{\partial^2 A(z_1, z_2)}{\partial z_i \partial z_j}\right|_{z_1=z_2=1}$ and $\mu_{jj} \triangleq \left.\frac{d^2 S_j(z)}{dz^2}\right|_{z=1}$, with $i, j = 1, 2$. Now we can calculate the mean system contents and the mean packet delay of both classes by taking the first derivatives of the respective pgf's for $z = 1$. We find

$$E[u_1] = \rho_1 + \frac{\lambda_{11}\mu_1 + \lambda_1^2 \mu_{11}}{2(1-\rho_1)}, \tag{12}$$

for the mean system contents of class 1 packets and

$$E[u_2] = \rho_2 + \frac{\rho_1 \lambda_2(\mu_2 - 1)}{1-\rho_1} + \frac{\lambda_{22}\mu_2}{2(1-\rho_T)} + \frac{\lambda_2^2 \mu_{22}}{2(1-\rho_T)(1-\rho_1)} + \frac{\lambda_{12}\mu_1}{1-\rho_T} + \frac{\lambda_2(\lambda_{11}\mu_1^2 + \lambda_1 \mu_{11})}{2(1-\rho_T)(1-\rho_1)}, \tag{13}$$

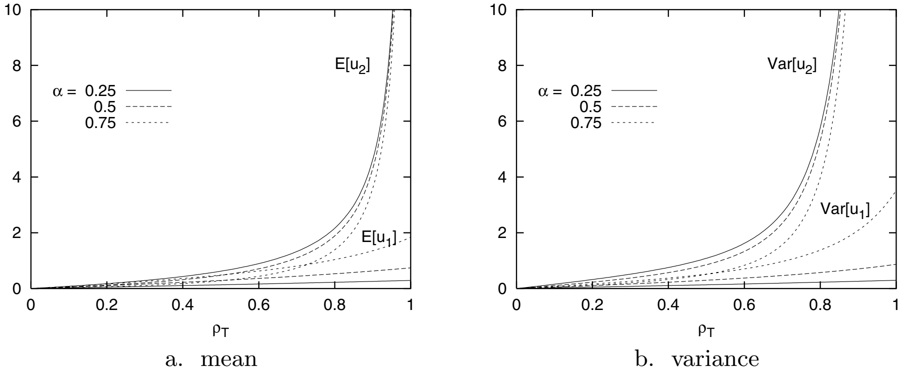
for the mean system contents of class 2 packets. The mean delay of both classes can also be found by taking the first derivatives of the respective pgf's for $z = 1$, and are given by $E[d_j] = E[u_j]/\lambda_j$. So, as expected, Little's law is satisfied.

In a similar way, expressions for the variance (and higher moments) can be calculated by taking the appropriate derivatives of the respective generating functions as well. These are nevertheless too elaborate to express them, but figures of the variance of system contents and packet delay of both classes will be shown in the next section.

6 Numerical Examples

In this section, we present some numerical examples. We assume the traffic of the two classes to be arriving according to a two-dimensional binomial process. Its two-dimensional pgf is given by $A(z_1, z_2) = (1 - \lambda_1(1 - z_1)/N - \lambda_2(1 - z_2)/N)^N$. The arrival rate of class j traffic is thus given by λ_j $(j = 1, 2)$. This arrival process occurs for instance at an output queue of a NxN output queueing switch/router fed by a Bernoulli process at the inlets. Notice also that if $N \to \infty$, the arrival process becomes a superposition of two independent Poisson streams. In the remainder of this section, we assume that $N = 16$. We furthermore denote the fraction of the high priority load in the total load by α, i.e., $\alpha = \rho_1/\rho_T$.

In Figure 1, the mean and variance of the system contents of class 1 and class 2 packets is shown as a function of the total load ρ_T, when service times of class 1 and class 2 packets are deterministically equal to 2 $(\mu_1 = \mu_2 = 2)$ and α is 0.25, 0.5 and 0.75 respectively. We clearly see the influence of the priority scheduling. The mean and variance of the system contents of class 1 packets remains low, even if the fraction of class 1 packets is high. The mean value and variance of the system contents of class 2 packets on the other hand is large, especially when the system is heavily loaded.



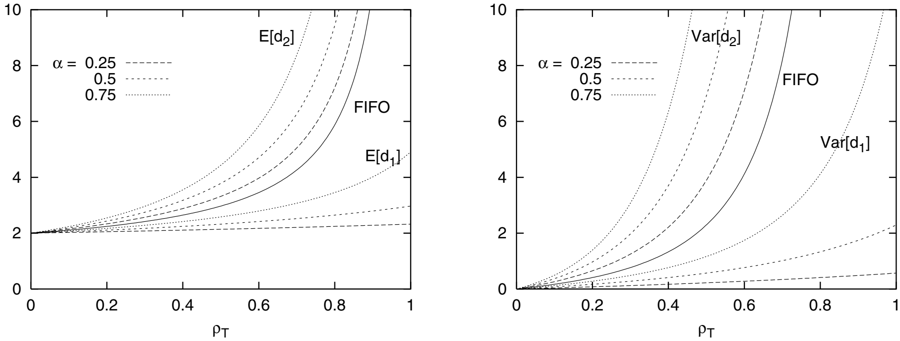
a. mean

b. variance

Fig. 1. Mean and variance of the system contents versus the total load

In Figure 2, the mean value and variance of the packet delay of class 1 and class 2 packets is shown as a function of the total load ρ_T, when service times of both classes are deterministically equal to 2, i.e., $\mu_j = 2$ $(j = 1, 2)$ and α is, as before, 0.25, 0.5 and 0.75 respectively. In order to compare with FIFO scheduling, we have also shown the mean value and variance of the packet delay in that case. Since, in this example, the service times of the class 1 and class 2 packets are equally distributed, the packet delay is then of course the same for class 1 and class 2 packets, and can thus be calculated as if there is only one class of packets arriving according to an arrival process with pgf $A(z, z)$.

This has already been analyzed, e.g., in [2]. The influence of priority scheduling on the packet delay becomes obvious from these figures: mean and variance of the delay of class 1 packets reduces significantly. The price to pay is of course a larger mean value and variance of the delay of class 2 packets. If this kind of traffic is not delay-sensitive, as assumed, this is not a too big a problem. Also, the smaller the fraction of high priority load in the overall traffic mix, the lower the mean and variance of the packet delay of both classes will be.



a. mean

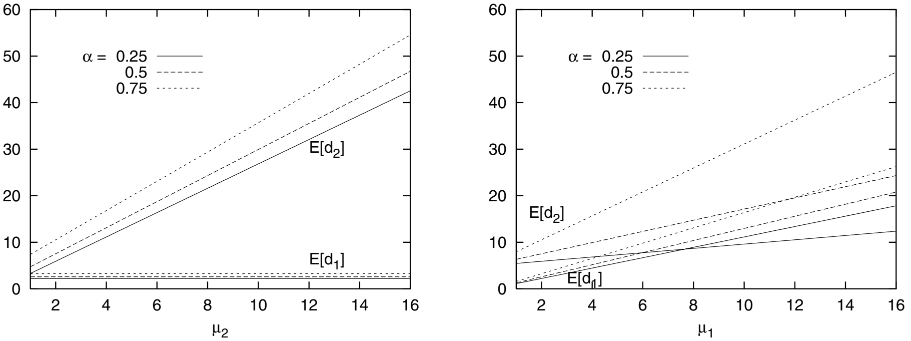
b. variance

Fig. 2. Mean and variance of the packet delay versus the total load

Finally, Figure 3a. (Figure 3b. respectively) shows the mean delay of high and low priority packets when service time of the packets are deterministic, as a function of the mean service time of the low priority packets (high priority packets respectively), i.e., μ_2 (μ_1 respectively), when $\mu_1 = 2$ ($\mu_2 = 2$ respectively) and $\rho_T = 0.75$. α is, as before, 0.25, 0.5 and 0.75. The figures show that the mean packet delay of high-priority packets is not influenced by the mean service time of class 2 packets, while it is proportionally increasing with the mean service time of class 1 packets (when the load of high and low priority packets is kept constant). The mean packet delay of class 2 packets on the other hand is proportionally increasing with the mean service time of class 2 packets and with the mean service time of class 1 packets. Because of the preemptive priority scheduling discipline, mean delay of high priority packets is only influenced by its own arrival and service process, while the mean delay of low priority packets is influenced by the arrival and service processes of both classes.

7 Conclusion

In this paper, we have analyzed a discrete-time $GI - G - 1$ queue with a preemptive resume priority scheduling and two priority classes. We have derived the joint generating function of the system contents of both classes and the generating functions of the delay of both classes. These pgf's are not explicitly found,



a. versus μ_2

b. versus μ_1

Fig. 3. Mean packet delay versus the mean service time of class 2 and class 1 packets

but we have proven that the moments of the distributions can be found explicitly in terms of the system parameters. We have shown the impact of priority scheduling on the performance characteristics by some numerical examples.

References

[1] H. Bruneel and B.G. Kim, *Discrete-time models for communication systems including ATM*, Kluwer Academic Publishers, Boston, 1993.

[2] H. Bruneel, *Performance of discrete-time queueing systems*, Computers and Operations Research, pp. 303-320, 1993.

[3] I. Cidon and R. Guérin, *On protective buffer policies*, Proceedings of Infocom '93 (San Francisco), pp. 1051-1058, 1993.

[4] L. Kleinrock, *Queueing systems volume II: Computer applications*, John Wiley & Sons, 1976.

[5] K. Liu, D.W. Petr, V.S. Frost, H. Zhu, C. Braun and W.L. Edwards, *Design and analysis of a bandwidth management framework for ATM-based broadband ISDN*, IEEE Communications Magazine, pp. 138-145, 1997.

[6] F. Machihara, *A bridge between preemptive and non-preemptive queueing models*, Performance Evaluation 23, pp. 93-106, 1995.

[7] R.G. Miller, *Priority queues*, Annals of Matematical. Statistics, pp. 86-103, 1960.

[8] S.P. Morgan, *Queueing disciplines and passive congestion control in byte-stream networks*, IEEE Transactions on Communications 39(7), pp. 1097-1106, 1991.

[9] H. Takagi, *Queueing analysis A foundation of Performance Evaluation Volume 1: Vacation and priority systems*, North-Holland, 1991.

[10] T. Takine and T. Hasegawa, *The workload in the MAP/G/1 queue with state-dependent services: its application to a queue with preemptive resume priority*, Commun. Statist.-Stochastic Models 10(1), pp. 183-204, 1994.

[11] J. Walraevens, B. Steyaert and H. Bruneel, Analysis of a preemptive resume priority buffer with general service times for the high priority class, Proceedings of the Africom 2001 Conference, Cape Town, May 28-30, 2001.

Analysis of the Discrete-Time $G^{(G)}/Geom/c$ Queueing Model

Sabine Wittevrongel[1], Herwig Bruneel[1], and Bart Vinck[2]

[1] Ghent University, Department of Telecommunications and Information Processing,
SMACS Research Group***, Sint-Pietersnieuwstraat 41, B-9000 Gent, Belgium
[2] Siemens AG – ICN M NT 18, Hofmannstraße 51, D-81359 München, Germany

Abstract. This paper presents the steady-state analysis of a discrete-time infinite-capacity multiserver queue with c servers and independent geometrically distributed service times. The arrival process is a batch renewal process, characterized by general independent batch interarrival times and general independent batch sizes. The analysis has been carried out by means of an analytical technique based on generating functions, complex analysis and contour integration. Expressions for the generating functions of the system contents during an arrival slot as well as during an arbitrary slot have been obtained. Also, the delay in case of a first-come-first-served queueing discipline has been analyzed.

1 Introduction

The analysis of discrete-time queueing models has received considerable attention in the scientific literature over the past years in view of its applicability in the study of many computer and communication systems in which time is slotted, see e.g. [1,2,3] and the references therein. In most of the existing studies of discrete-time multiserver queueing models, however, the service times of customers are assumed to be constant, equal to one slot (see [4,5,6]) or multiple slots ([7]). On the other hand, very little seems to have been done on discrete-time multiserver queues with random service times. A multiserver queueing system with geometric service times and a general independent arrival process is analyzed in [8]. Geometric service times are also considered in [9,10,11], while [12] and [13] deal with general service times, but only for the single-server case.

In this paper, we present an analytical technique for the analysis of discrete-time multiserver queues with geometric service times and a batch renewal arrival process. This process is characterized by a sequence of independent and identically distributed (i.i.d.) batch interarrival times and a sequence of i.i.d. batch sizes, and can be used to model both first- and second-order correlation characteristics of a traffic stream ([14]). As far as know to the authors, an analysis of the considered queueing model has never been reported on in the literature.

The remainder of the paper is organized as follows. The assumptions of the queueing model under study and some basic terminology are given in Sect. 2.

[***] SMACS: Stochastic Modeling and Analysis of Communication Systems

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 757–768, 2002.

Next, the analysis of the queueing model is carried out and expressions are derived for the generating functions of the system contents during an arrival slot (Sect. 3), the system contents during an arbitrary slot (Sect. 4) and the waiting time and delay in case of a first-come-first-served queueing discipline (Sect. 5).

2 Model Description

In this paper, we consider a discrete-time buffer system with an infinite waiting room for customers and c servers. It is assumed the system has a clock such that time is divided in fixed-length *slots* s_j $(j = 1, 2, \dots)$, chronologically indexed, and separated by *slot boundaries* t_j, where $s_j \triangleq [t_j, t_{j+1})$.

Arrivals to the system occur solely on slot boundaries. Slot boundaries with arrivals are called *arrival instants*; the slot following an arrival instant is called an *arrival slot*. We denote the kth arrival instant by τ_k and the arrival slot following τ_k by I_k $(k = 1, 2, \dots)$. The time interval (expressed in slots) between two successive arrival instants is referred to as the *interarrival time* and is denoted by the symbol A. Specifically, A_k stands for the interarrival time starting at τ_k. The interarrival times are assumed to be independent and identically distributed (i.i.d.) random variables with common probability mass function (PMF) $a(n) \triangleq \text{Prob}[A_k = n]$ $(n = 1, 2, \dots)$, and probability generating function (PGF) $A(z)$.

At a given arrival instant, several customers may enter the system (batch arrivals). In particular, the random variable for the number of customers arriving at τ_k is denoted by B_k and is called the *batch size* for the arrival instant τ_k. The batch sizes are assumed to be i.i.d. random variables. Their common PGF is denoted by $B(z)$. All A_k's and B_k's are also mutually statistically independent.

Customers are queued for service according to a first-come-first-served discipline (FCFS), and receive service from any of the c servers. Hereby the order in which simultaneously arriving customers are queued for service is irrelevant for the analysis. Service can start solely at slot boundaries and always takes a positive integer number of full slots. A customer can be taken in service as soon as he arrives, provided, of course, there is a server available. After service completion customers leave the system immediately. Hence, the departures from the system also occur solely at slot boundaries. Note that the assumption of a FCFS queueing discipline has no influence on the distribution of the buffer contents.

The number of slots it takes to serve the lth customer is called the lth *service time*, and is denoted by D_l. The service times are assumed to be i.i.d. and geometrically distributed with parameter $1 - \sigma$ $(0 < \sigma \leq 1)$, i.e., with PGF

$$D(z) = \frac{\sigma z}{1 - (1 - \sigma) z} \ . \tag{1}$$

3 System Contents during an Arrival Slot

During any slot the number of customers in the system, referred to as the *system contents*, remains constant and therefore is well-defined. For any arrival slot I_k, let U_k be the system contents during I_k. Due to the memoryless nature of the service-time distribution, the random variables $\{U_k \mid k = 1, 2, \dots\}$ form a Markov chain. Let $U_k(z)$ be the PGF of U_k. Under the assumption that the buffer system, on the average, receives less work than it can handle, i.e., $B'(1) < c\sigma A'(1)$, the system will, for large k, tend towards an equilibrium, where all U_k's have a common distribution with PGF $U(z)$.

Now, let us consider an arbitrary pair of two consecutive arrival slots I_k and I_{k+1}, and the time interval in between them. We define the random variable V_p ($p = 0, 1, \dots, A_k$) as the number of customers present in the system in slot I_k and still present in the system during the slot $s^*_p = [t^*_p, t^*_{p+1})$, i.e., the pth slot after I_k (see Fig. 1). For $p < A_k$, V_p represents the actual system contents during s^*_p, while for $p = A_k$, we have $s^*_{A_k} = I_{k+1}$ and the system contents in this slot is

$$U_{k+1} = V_{A_k} + B_{k+1} \ . \tag{2}$$

The PMF of the random variable V_p is denoted by $v_p(n) \triangleq \mathrm{Prob}[V_p = n]$ ($n = 0, 1, \dots$), and its PGF by $V_p(z)$. It is clear that the distribution of V_p depends only on the distribution of $V_0 = U_k$ and the distribution of the total number of departures at the slot boundaries in the interval $]\tau_k, t^*_p]$. Between the random variables related to consecutive slots the following relation holds :

$$V_p = V_{p-1} - R_p \ , \quad p = 1, 2, \dots, A_k \ , \tag{3}$$

where the random variable R_p indicates the number of departures at t^*_p (see Fig. 1). Since the service times are geometrically distributed with parameter $(1-\sigma)$, the number of departures at t^*_p ($p = 1, 2, \dots, A_k$) has a binomial distribution with parameters $r_p = \min(c, V_{p-1})$ and σ, r_p being the number of servers that are occupied during the preceding slot s^*_{p-1}. In terms of the conditional PGF

$$R(z \mid n) \triangleq \sum_{m=0}^{\min(c,n)} \mathrm{Prob}[R_p = m \mid V_{p-1} = n] \ z^m \ , \tag{4}$$

we have that

$$R(z \mid n) = \begin{cases} (1-\sigma+\sigma z)^n \triangleq [M(z)]^n \ , & 0 \le n \le c-1 \ ; \\ (1-\sigma+\sigma z)^c \triangleq [M(z)]^c \ , & c \le n \ . \end{cases} \tag{5}$$

Equations (3)–(5) then yield

$$\begin{aligned} V_p(z) &= \sum_{n=0}^{\infty} \sum_{m=0}^{\min(c,n)} \mathrm{Prob}[V_{p-1} = n] \ \mathrm{Prob}[R_p = m \mid V_{p-1} = n] \ z^{n-m} \\ &= [M(1/z)]^c \, V_{p-1}(z) + \sum_{n=0}^{c-1} v_{p-1}(n) \, F_n(z) \ , \quad p = 1, 2, \dots, A_k \ , \end{aligned} \tag{6}$$

where $\{F_n(z) \mid n = 0, 1, \dots, c-1\}$ is a family of rational functions in z :

$$F_n(z) \triangleq z^n \ \left([M(1/z)]^n - [M(1/z)]^c\right) \ , \quad n = 0, 1, \dots, c-1 \ . \tag{7}$$

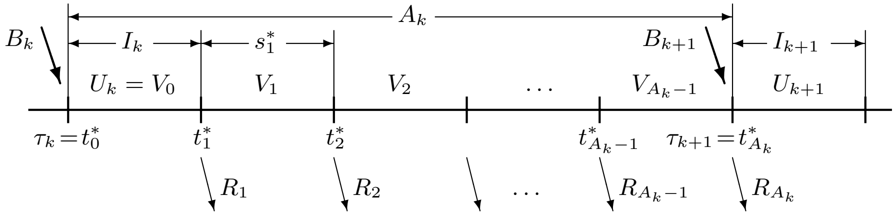


Fig. 1. System contents during an arrival slot

Next, applying (6) repeatedly, and taking into account that $V_0 = U_k$, or equivalently $V_0(z) = U_k(z)$, we get that

$$V_p(z) = [M(1/z)]^{p\,c}\, U_k(z) + \sum_{n=0}^{c-1} F_n(z) \sum_{m=0}^{p-1} v_m(n)\, [M(1/z)]^{(p-m-1)\,c} \ . \tag{8}$$

In view of (2), and since B_{k+1} is independent of V_{A_k}, we have

$$U_{k+1}(z\,|\,p) \triangleq E\big[z^{U_{k+1}}\,|\,A_k = p\big] = B(z)\, V_p(z) \ . \tag{9}$$

From (8)–(9), the unconditional PGF $U_{k+1}(z)$ of U_{k+1} is then derived as

$$\begin{aligned} U_{k+1}(z) &= B(z)\, A([M(1/z)]^c)\ U_k(z) \\ &\quad + B(z) \sum_{n=0}^{c-1} F_n(z) \sum_{p=1}^{\infty} a(p) \sum_{m=0}^{p-1} v_m(n)\, [M(1/z)]^{(p-m-1)\,c} \ . \end{aligned} \tag{10}$$

Under the assumption of equilibrium, both $U_k(z)$ and $U_{k+1}(z)$ can be substituted by $U(z)$. Solving for $U(z)$, we obtain

$$U(z) = \frac{B(z)}{1 - B(z)\, A([M(1/z)]^c)} \sum_{n=0}^{c-1} F_n(z)\, J_n(z) \ , \tag{11}$$

where the functions $J_n(z)$ are defined as

$$J_n(z) \triangleq \sum_{p=1}^{\infty} a(p)\, [M(1/z)]^{(p-1)\,c} \sum_{m=0}^{p-1} v_m(n)\, [M(1/z)]^{-m\,c} \ . \tag{12}$$

The right-hand side of (11) contains, through the functions $J_n(z)$, an infinite number of unknown coefficients $\{v_m(n)\,|\,m = 0, 1, \ldots\,; n = 0, 1, \ldots\,, c{-}1\}$. Based on these probabilities, we define a family of c functions by their Taylor series expansions around $z = 0$:

$$H_n(z) \triangleq \sum_{m=0}^{\infty} v_m(n)\, z^m \ , \quad n = 0, 1, \ldots\,, c-1 \ . \tag{13}$$

These series converge at least for all z with $|z| < 1$, since the sum of all of the coefficients are bounded by 1. Since $\sigma > 0$, all customers present in the buffer during I_k will eventually leave the system, so that $\lim_{m\to\infty} v_m(n) = \delta(n)$, where

$$\delta(n) \triangleq \begin{cases} 0 \ , & \text{if } n \neq 0 \ ; \\ 1 \ , & \text{if } n = 0 \ . \end{cases} \tag{14}$$

Hence, we can expect the series $H_n(z)$, for all $n \in \{1, 2, \dots, c-1\}$, to converge in a region that contains the unit disk, including its edge. On the other hand, $H_0(z)$ apparently has a pole in $z = 1$. The coefficient $v_m(n)$ can be written as the residue of the complex function $H_n(\zeta)\,\zeta^{-m-1}$ at $\zeta = 0$, so that

$$v_m(n) = \frac{1}{2\pi i} \oint_L \frac{H_n(\zeta)}{\zeta^{m+1}}\, d\zeta \ , \tag{15}$$

where i indicates the imaginary unit and L, for the time being, is an arbitrary closed contour around the origin $\zeta = 0$ in the complex ζ-plane, but not around any other singularity of $H_n(\zeta)$. After substitution in (12), under the assumption that a proper choice is made for the contour L and for proper z, the summations over p and m on the right-hand side of (12) can be brought behind the integration operator and summed. As a result, we get the following expression for $J_n(z)$:

$$J_n(z) = \frac{1}{2\pi i} \oint_L \frac{H_n(\zeta)}{\zeta\,[M(1/z)]^c - 1}\, \left[A([M(1/z)]^c) - A(1/\zeta)\right]\, d\zeta \ , \tag{16}$$

for all z such that $|M(1/z)|^c < \mathcal{R}_A$, where the notation $\mathcal{R}_X$ stands for the radius of convergence of the Taylor series expansion of the function $X(z)$ around the origin $z = 0$. This range for z contains at least the entire complex z-plane outside the unit circle and the unit circle itself. For ζ, on the other hand, a "proper" choice for the contour L means that

$$(\forall\zeta \in L)\, (1/\,|\zeta| < \mathcal{R}_A \text{ and } |\zeta| < \mathcal{R}_{H_n}) \ . \tag{17}$$

For all $n \in \{0, 1, \dots, c-1\}$, we can choose for L an arbitrary circle with center in $\zeta = 0$ and radius a, where $1/\mathcal{R}_A < a < 1$, i.e., a circle smaller than the unit circle but still around all the singularities of $A(1/\zeta)$.

In view of Cauchy's residue theorem ([15]), $J_n(z)$ can be obtained as a sum of integrals over small contours around the singularities of the integrand inside the contour L. These singularities are given by the set $\mathcal{S}_A^{-1}$, i.e., the set of singular points of the function $A(1/\zeta)$. They all have a modulus smaller than $1/\mathcal{R}_A$ and therefore lie within the contour L, so that

$$J_n(z) = \sum_{\alpha\in\mathcal{S}_A^{-1}} \frac{1}{2\pi i} \oint_{L_\alpha} \frac{H_n(\zeta)}{1 - \zeta\,[M(1/z)]^c}\, A(1/\zeta)\, d\zeta \ . \tag{18}$$

Note that there is no contribution of the term with $A([M(1/z]^c)$ because it remains regular for all $\zeta \in \mathcal{S}_A^{-1}$. In (18), L_α is a "sufficiently" small contour around $\alpha \in \mathcal{S}_A^{-1}$, i.e., inside the circle $C(0, 1/\mathcal{R}_A)$ and not around any other singularity of $A(1/\zeta)$. Because in (16), z solely occurs under the form $[M(1/z)]^c$, it is expedient to make a change of variable towards

$$u = u(z) \triangleq [M(1/z)]^{-1} \equiv D^{-1}(z) \Leftrightarrow z = D(u) \ , \tag{19}$$

so that we get for $J_n(z)$:

$$J_n(z) \triangleq \tilde{J}_n(u^c) = u^c \sum_{\alpha \in \mathcal{S}_A^{-1}} \tilde{J}_{n,\alpha}(u^c) \ , \tag{20}$$

where

$$\tilde{J}_{n,\alpha}(x) \triangleq \frac{1}{2\pi i} \oint_{L_\alpha} \frac{H_n(\zeta)}{x - \zeta} A(1/\zeta)\, d\zeta \ . \tag{21}$$

For any x outside L_α, the contribution $\tilde{J}_{n,\alpha}(x)$ for any $\alpha \in \mathcal{S}_A^{-1}$, is the integral along a *finite* contour of an integrand that remains regular along that contour, and therefore is a well-defined complex number. Further, the integrand is an analytical function of x for all x outside of L_α, so that also $\tilde{J}_{n,\alpha}(x)$ is analytical outside of L_α. Finally, since L_α can be chosen arbitrarily small, $\tilde{J}_{n,\alpha}(x)$ is a regular analytical function for all $x \neq \alpha$. At $x = \alpha$, however, $\tilde{J}_{n,\alpha}(x)$ has a singularity, exactly of the same type as for $A(1/\zeta)$ at $\zeta = \alpha$. Also, $\lim_{x \to \infty} \tilde{J}_{n,\alpha}(x) = 0$, so that the Laurent series expansion around $x = \alpha$ for $\tilde{J}_{n,\alpha}(x)$ does not contain terms with positive powers of $(x - \alpha)$. For an essential singularity, little more can be said because the functions $H_n(\zeta)$ are unknown. However, if $\alpha \in \mathcal{S}_A^{-1}$ is a pole with multiplicity m, $\tilde{J}_{n,\alpha}(x)$ is of the following form :

$$\tilde{J}_{n,\alpha}(x) = \frac{\Lambda_{-m}}{(x-\alpha)^m} + \frac{\Lambda_{-m+1}}{(x-\alpha)^{m-1}} + \ldots + \frac{\Lambda_{-1}}{x - \alpha} \ , \tag{22}$$

where

$$\Lambda_{-k} = \frac{1}{(m-k)!} \lim_{\zeta \to \alpha} \frac{\mathrm{d}^{m-k}}{\mathrm{d}\zeta^{m-k}} \left[H_n(\zeta)\, A(1/\zeta)\, (\zeta - \alpha)^m \right] \ , \ \ k = 1, 2, \ldots, m \ . \tag{23}$$

The Λ_{-k}'s still depend on $H_n(\alpha), H_n'(\alpha), \ldots, H_n^{(m-k)}(\alpha)$ and therefore are to be considered as unknown quantities. However, further analysis will show that it is not necessary to determine these coefficients.

From this point on we assume that $A(z)$ is a rational function. When $A(z)$ is rational, all the singularities of $A(1/z)$ are poles and all of them give contributions of the form of (22). Let us write

$$A(1/z) \triangleq \frac{P_A(z)}{Q_A(z)} \ ; \tag{24}$$

$$Q_A(z) \triangleq \prod_{\alpha \in \mathcal{S}_A^{-1}} (z - \alpha)^{m_\alpha} \ , \tag{25}$$

where m_α indicates the multiplicity of $\alpha \in \mathcal{S}_A^{-1}$ and $P_A(z)$ is a polynomial function with $\deg P_A < \deg Q_A$, since $\lim_{z\to\infty} A(1/z) = 0$. Summing over the contributions of all poles, we can easily see that

$$\sum_{\alpha\in\mathcal{S}_A^{-1}} \tilde{J}_{n,\alpha}(x) = \frac{P_{U,n}(x)}{Q_A(x)} \ , \quad n = 0, 1, \dots, c-1 \ , \tag{26}$$

with $P_{U,n}(x)$ a yet unknown polynomial function of degree $\deg P_{U,n} = \deg Q_A - 1$. Hence, (20) becomes

$$J_n(z) = \tilde{J}_n(u^c) = u^c \, \frac{P_{U,n}(u^c)}{Q_A(u^c)} \ . \tag{27}$$

Next, from (11) and (27), in view of

$$F_n(z) \triangleq \tilde{F}_n(u) = [D(u)]^n \left(u^{-n} - u^{-c}\right) \ , \quad n = 0, 1, \dots, c-1 \ , \tag{28}$$

we readily get for $U(z)$:

$$U(z) = \tilde{U}(u) = \frac{B(D(u))\, \tilde{P}_U(u)}{[1-(1-\sigma)u]^{c-1}\,[Q_A(u^c) - B(D(u))\,P_A(u^c)]} \ , \tag{29}$$

where

$$\tilde{P}_U(u) \triangleq \sum_{n=0}^{c-1} \sigma^n \left(u^c - u^n\right) [1-(1-\sigma)u]^{c-n-1} \, P_{U,n}(u^c) \ , \tag{30}$$

which is the sole unknown function on the right-hand side of (29). $\tilde{P}_U(u)$ is a polynomial function in u of degree $(c \deg Q_A + c - 1)$, though, the coefficients have only $c \deg Q_A$ degrees of freedom, so that $c \deg Q_A$ linearly independent conditions on the coefficients suffice to determine the polynomial $\tilde{P}_U(u)$ completely. These conditions are obtained by invoking the analyticity of the PGF $U(z)$ in the unit disk, i.e., the analyticity of $\tilde{U}(u)$ in the image of the unit disk under the transformation $z \to u = D^{-1}(z)$ and more particularily in the zeros of the denominator of (29) in that area. An application of Rouché's theorem ([16]) shows that the factor

$$Q_U(z) \triangleq Q_A([M(1/z)]^{-c}) - B(z)\, P_A([M(1/z)]^{-c}) \tag{31}$$

has as many zeros within the unit disk as $Q_A([M(1/z)]^{-c})$, counting multiple zeros several times. This number of zeros is precisely $c \deg Q_A$. One of these zeros of $Q_U(z)$ is 1, all other zeros lie strictly within the unit circle. Therefore, also

$$\tilde{Q}_U(u) = Q_A(u^c) - B(D(u))\, P_A(u^c) \tag{32}$$

has $c \deg Q_A$ zeros within the image of the unit disk under the mapping $z \to D^{-1}(z)$. One of these zeros is 1, where $\tilde{P}_U(u)$ vanishes regardless the coefficients of the polynomials $P_{U,n}$. Let us denote the set of the $c \deg Q_A - 1$ other zeros by $Q_U = \{u : \tilde{Q}_U(u) = 0 \text{ and } |D(u)| < 1\}$. In these points, $\tilde{U}(u)$ must remain regular, and therefore they have to be zeros of $\tilde{P}_U(u)$ as well, with at least the same multiplicity, so that $c \deg Q_A - 1$ linear equations in the $c \deg Q_A$ unknown coefficients are obtained. An additional equation follows from the normalization condition $\tilde{U}(1) = 1$. Hence, we have a set of $c \deg Q_A$ linearly independent conditions on the $c \deg Q_A$ unknown coefficients, so that the polynomial $\tilde{P}_U(u)$ can be determined completely. Note that $\tilde{P}_U(u)$ will be of the form

$$\tilde{P}_U(u) = \Psi(u) \prod_{\beta \in Q_U} (u - \beta)^{m_\beta} \ , \tag{33}$$

where $\Psi(u)$ is a polynomial of degree c, and m_β indicates the multiplicity of the zero β of $\tilde{Q}_U(u)$ (and $\tilde{P}_U(u)$). Thus finally, $U(z)$ is completely expressed in terms of known quantities only, i.e., the PGFs $A(z)$ and $B(z)$, the parameter σ, the $\deg Q_A$ poles of the function $A(1/z)$, and the $c \deg Q_A - 1$ solutions of $Q_U(z) = 0$, or, equivalently, of the characteristic equation $1 = B(z)\, A([M(1/z)]^c)$ strictly inside the unit disk.

4 System Contents during an Arbitrary Slot

Under the assumption that the buffer system has reached a stochastic equilibrium, we now consider an arbitrary slot s and we let N denote the system contents during slot s (see Fig. 2). The start of s is indicated by t. Also, let τ be the preceding arrival instant (if t is an arrival instant, let $\tau = t$), let I be the arrival slot starting at τ and let U be the system contents during I.

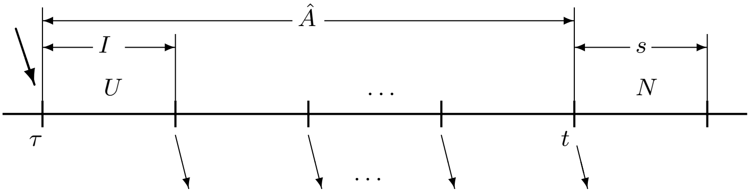


Fig. 2. System contents during an arbitrary slot

It is clear that the distribution of N depends only on the distribution of U and the distribution of the total number of departures at the slot boundaries in the interval $]\tau, t]$. Let the random variable $\hat{A}$, with PMF $\hat{a}(n)$ $(n = 0, 1, \dots)$ and PGF $\hat{A}(z)$, be the number of slots between τ and t, then $\hat{A}(z)$ is given by (see e.g. [2]) :

$$\hat{A}(z) = \frac{A(z) - 1}{A'(1)\,(z - 1)} \ , \tag{34}$$

which implies that the singularities of $\hat{A}(z)$ and $A(z)$ coincide. By applying the method introduced in Sect. 3, we can derive the PGF $N(z)$ of N in terms of the PGF $U(z)$ of U as

$$N(z) = U(z)\,\hat{A}([M(1/z)]^c) + \sum_{n=0}^{c-1} F_n(z)\,K_n(z) \ , \tag{35}$$

where the functions $K_n(z)$ are defined as

$$K_n(z) \triangleq \sum_{p=0}^{\infty} \hat{a}(p) \sum_{m=0}^{p-1} v_m(n)\,[M(1/z)]^{(p-m-1)\,c} \ , \quad n = 0, 1, \ldots, c-1 \ . \tag{36}$$

Again, we now assume that $A(z)$ is rational. In this case, $\hat{A}(z)$ is a rational function as well. Also $A(1/z)$ and $\hat{A}(1/z)$ have exactly the same set of poles, i.e.,

$$\hat{A}(1/z) = \frac{P_{\hat{A}}(z)}{Q_A(z)} \ , \tag{37}$$

where $Q_A(z)$ is given by (25), and $P_{\hat{A}}(z)$ is a polynomial function with $\deg P_{\hat{A}} = \deg Q_A$, since $\lim_{z\to\infty} \hat{A}(1/z) = 1/A'(1)$. In a similar way as explained for the functions $J_n(z)$ in the previous section, it can be shown that the functions $K_n(z)$ are of the following form :

$$K_n(z) = \tilde{K}_n(u^c) = u^c\,\frac{P_{N,n}(u^c)}{Q_A(u^c)} \ , \tag{38}$$

where the $P_{N,n}(x)$ are unknown polynomials in x of degree $\deg P_{N,n} = \deg Q_A - 1$, and again we have made a change of variable according to (19). Finally, combining (28), (35) and (38), we get the PGF $N(z)$ as

$$N(z) = \tilde{N}(u) = \tilde{U}(u)\,\frac{P_{\hat{A}}(u^c)}{Q_A(u^c)} + \frac{\tilde{P}_N(u)}{[1-(1-\sigma)u]^{c-1}\,Q_A(u^c)} \ , \tag{39}$$

where

$$\tilde{P}_N(u) \triangleq \sum_{n=0}^{c-1} \sigma^n\,(u^c - u^n)\,[1-(1-\sigma)u]^{c-n-1}\,P_{N,n}(u^c) \tag{40}$$

is a polynomial in u of degree $(c\deg Q_A + c - 1)$, with $c\deg Q_A$ unknown coefficients. Again, the required conditions for the determination of $\tilde{P}_N(u)$ are obtained by imposing the analyticity of the PGF $N(z)$ everywhere in the unit disk, or equivalently, the analyticity of $\tilde{N}(u)$ in the image of the unit disk under the transformation $z \to u = D^{-1}(z)$. As before, note that $Q_A([M(1/z)]^{-c})$ has exactly $c\deg Q_A$ zeros within the unit disk of the complex z-plane, and hence, $Q_A(u^c)$ has $c\deg Q_A$ zeros within the image of the unit disk under the transformation $z \to D^{-1}(z)$. In order for $\tilde{N}(u)$ to remain regular in these points, they also have to be zeros of the numerator of $\tilde{N}(u)$, with at least the same multiplicity. This gives us a set of $c\deg Q_A$ linearly independent equations in the $c\deg Q_A$ unknown coefficients of $\tilde{P}_N(u)$, so that $\tilde{P}_N(u)$, and hence $\tilde{N}(u)$, is completely determined.

5 Waiting Time and Delay

We now derive the PGF of the waiting time W of an arbitrary customer, denoted by C, arriving in the system when equilibrium has established itself. Let us denote the arrival instant of C by τ_a. Owing to the FCFS queueing discipline, the waiting time W of customer C depends on the total number of customers present in the buffer system during the arrival slot of C which have priority over customer C to be taken into service. This number of customers is a random variable, denoted by T, which is the sum of the number of customers staying in the system at τ_a (i.e., the number of customers present in the system both during the slot before and the slot after τ_a), and the number of customers arriving at τ_a (simultaneously with C) and being queued for service ahead of C. In stochastic equilibrium, the first component has PGF

$$V(z) = \frac{U(z)}{B(z)} \ , \tag{41}$$

and the second component has PGF (see e.g. [2])

$$\hat{B}(z) = \frac{B(z)-1}{B'(1)\,(z-1)} \ , \tag{42}$$

since C is an arbitrary customer, and hence in an arbitrary position within the batch of customers arriving at τ_a. Clearly, the two components of T are independent random variables, so that the PGF of T is given by

$$T(z) = V(z)\,\hat{B}(z) \ . \tag{43}$$

Let us consider the conditional probabilities

$$g(n\,|\,k) \triangleq \text{Prob}[W = n\,|\,T = k] \quad , \quad n \geq 0 \quad , \quad k \geq 0 \ . \tag{44}$$

The way in which the distribution of W depends on the distribution of T is determined uniquely by the departure process. When $T < c$, customer C will be taken into service as soon as he arrives and his waiting time will be zero. Hence,

$$g(n\,|\,k) = \delta(n) \quad , \quad 0 \leq k < c \ . \tag{45}$$

For $T \geq c$, on the other hand, the waiting time of C cannot be zero. In that case, however, the following recurrence relationship holds :

$$\begin{cases} g(0\,|\,k) = 0 \ , & k \geq c \ ; \\ g(n\,|\,k) = \sum\limits_{l=0}^{c} r(l)\,g(n-1\,|\,k-l) \ , & k \geq c \ , \quad n \geq 1 \ . \end{cases} \tag{46}$$

Here $r(l)$ ($l = 0, 1, \ldots, c$) is the PMF of the number of departures at a slot boundary when a customer is waiting and hence all the servers are occupied.

Due to the geometric service-time distribution, this number of departures has a binomial distribution with parameters c and σ, i.e., a PGF

$$R(z) = (1 - \sigma + \sigma z)^c \ . \tag{47}$$

The recurrence relationship (46) can be transformed in an algebraic one for

$$G(x \,|\, y) \triangleq \sum_{n=0}^{\infty} \sum_{k=0}^{\infty} g(n \,|\, k)\, x^n\, y^k \ . \tag{48}$$

More specifically, we get

$$\begin{aligned} G(x \,|\, y) &= \sum_{k=0}^{c-1} y^k + \sum_{n=1}^{\infty} \sum_{k=c}^{\infty} \sum_{l=0}^{c} r(l)\, g(n-1 \,|\, k-l)\, x^n\, y^k \\ &= \frac{1-y^c}{1-y} + x \left[y^c \sum_{l=0}^{c} r(l)\, \frac{1-y^l}{1-y} + R(y) \left(G(x \,|\, y) - \frac{1-y^c}{1-y} \right) \right] \ . \end{aligned} \tag{49}$$

Solving for $G(x \,|\, y)$, we find

$$G(x \,|\, y) = \frac{1}{1-y} \left(1 - y^c \, \frac{1-x}{1 - x\, R(y)} \right) \ , \tag{50}$$

which is a valid expression for $G(x \,|\, y)$ wherever $|x| < 1/\,|R(y)|$ in view of the factor $[1 - x\, R(y)]$ in the denominator of (50).

Together with the distribution of T (equation (43)), this result suffices to obtain the PGF $W(z)$ of the waiting time W of an arbitrary customer. Indeed, with $t(k) \triangleq \mathrm{Prob}[T = k]$ $(k = 0, 1, \ldots)$,

$$W(z) = \sum_{n=0}^{\infty} \sum_{k=0}^{\infty} g(n \,|\, k)\, t(k)\, z^n = \frac{1}{2\pi i} \oint_L \frac{T(\zeta)}{\zeta}\, G\!\left(z \,\middle|\, \frac{1}{\zeta} \right) d\zeta \ , \tag{51}$$

with L a contour around the origin $\zeta = 0$, but not around any singularity of $T(\zeta)$ and such that for all $\zeta \in L$, the sum over k and n converges, i.e., for all $\zeta \in L$, $|z| < 1/\,|R(1/\zeta)|$. The integrand is given by

$$\frac{T(\zeta)}{\zeta}\, G\!\left(z \,\middle|\, \frac{1}{\zeta} \right) = \frac{T(\zeta)}{\zeta - 1} \left[1 - \frac{1-z}{\zeta^c - z\, \tilde{R}(\zeta)} \right] \ , \tag{52}$$

where $\tilde{R}(\zeta) \triangleq \zeta^c\, R(1/\zeta)$ is a polynomial in ζ of degree c. From the Theorem of Rouché, it follows that the numerator factor $\zeta^c - z\, \tilde{R}(\zeta)$ has c zeros inside the unit circle, counting multiple zeros several times. Let us denote these zeros by $\beta_j(z)$ $(j = 0, 1, \ldots, c-1)$. They are given by

$$\beta_j(z) = - \frac{\sigma}{1 - \sigma - |z|^{-1/c} \exp[\frac{i}{c}\,(-\arg z + 2\pi j)]} \ , \quad j = 0, 1, \ldots, c-1 \ , \tag{53}$$

where $\beta_0(1) = 1$. Apparently, for any $z \not= 0$ all$\beta_j(z)$ are different so that the contour integral in (51) can be obtained as the sum over the residues of the integrand in the simple poles of the integrand. The result reads

$$W(z) = \frac{z-1}{z} \sum_{j=0}^{c-1} \frac{T(\beta_j(z))}{[\beta_j(z)-1]\beta_j(z)^{c-2}R'(1/\beta_j(z))} \ . \tag{54}$$

Finally, the PGF of the complete system time or delay S of C, in which also the service time is included, is then obtained as

$$S(z) = W(z)\, D(z) \ . \tag{55}$$

References

1. Hunter, J.J.: Mathematical techniques of applied probability, Vol. 2, Discrete time models: techniques and applications. Academic Press, New York (1983)
2. Bruneel, H., Kim, B.G.: Discrete-time models for communication systems including ATM. Kluwer Academic Publishers, Boston (1993)
3. Woodward, M.E.: Communication and computer networks: modelling with discrete-time queues. Pentech Press, London (1993)
4. Chu, W.W.: Buffer behavior for Poisson arrivals and multiple synchronous constant outputs. IEEE Trans. Comput. **19** (1970) 530–534
5. Li, S.-Q.: A general solution technique for discrete queueing analysis of multimedia traffic on ATM. IEEE Trans. Commun. **39** (1991) 1115–1132
6. Bruneel, H., Steyaert, B., Desmet, E., Petit, G.H.: An analytical technique for the derivation of the delay performance of ATM switches with multiserver output queues. Int. J. Digital and Analog Commun. Syst. **5** (1992) 193–201
7. Bruneel, H., Wuyts, I.: Analysis of discrete-time multiserver queueing models with constant service times. Oper. Res. Lett. **15** (1994) 231–236
8. Rubin, I., Zhang, Z.: Message delay and queue-size analysis for circuit-switched TDMA systems. IEEE Trans. Commun. **39** (1991) 905–914
9. Hsu, J.: Buffer behavior with Poisson arrival and geometric output processes. IEEE Trans. Commun. **22** (1974) 1940–1941
10. Vinck, B., Bruneel, H.: Analyzing the discrete-time $G^{(G)}$/Geo/1 queue using complex contour integration. Queue. Syst. **18** (1994) 47–67
11. Gao, P., Wittevrongel, S., Bruneel, H., Zhang, S.: Analysis of discrete-time buffers with geometric service times and correlated input traffic. In: Proc. High Performance Computing Symposium, HPC 2001. Seattle (April 2001) 251–256
12. Briem, U., Theimer, T.H., Kröner, H.: A general discrete-time queueing model: analysis and applications. In: Proc. ITC 13, Vol. Teletraffic and Datatraffic in a Period of Change. Copenhagen (June 1991) 13–19
13. Murata, M., Miyahara, H.: An analytic solution of the waiting time distribution for the discrete-time GI/G/1 queue. Perf. Eval. **13** (1991) 87–95
14. Kouvatsos, D., Fretwell, R.: Batch renewal process: exact model of traffic correlation. In: High-Speed Networking for Multimedia Applications. Kluwer Academic Publishers, Boston (1996) 285–304
15. González, M.O.: Classical complex analysis. Marcel Dekker Inc., New York (1992)
16. Kleinrock, L.: Queueing systems, Vol. I: Theory. John Wiley & Sons, New York (1975)

On a Theory of Interacting Queues

Alexander Stepanenko[1], Costas C. Constantinou[1], Theodoros N. Arvanitis[1], and Kevin Baughan[2]

[1] School of Electrical, Electronic and Computer Engineering, University of Birmingham, Edgbaston, B15 2TT, UK
[2] Nortel Networks, Maidenhead Office Park, Westacott Way, Maidenhead, Berkshire, SL6 3QH, UK

Abstract. We present a possible way to extend queuing theory to account for interactions between adjacent queues in a packet-switched network. The interaction between queues arises because of the influence of the routing protocol on each switching decision and the stochastic nature of packet lengths and inter-arrival times.
Both the methodology and the analysis tools are adaptations of methods of statistical mechanics and are presented in outline here. The justification for their use lies in experimental evidence given in [1,2,3] that aggregate, core-network IP traffic exhibits quasi-Markovian properties.
In this paper, we focus on the interaction between pairs of queues, either in a cascaded arrangement, or connected to the same switching fabric, in the presence of an idealised routing protocol.

1 Introduction

Next generation telecommunication networks are likely to rely on an IP core network infrastructure. As a consequence, unlike today's Internet, future networks will be subject to much more demanding requirements. In order to provide operational guarantees, network owners need tools which enable the dimensioning of the core of packet-switched networks. It is customary that network design is based mainly on simulations. Replacing theory by simulations is non-tenable due to the sheer size of such networks.

A theory of traffic in packet-switched networks must be capable of predicting a number of quantities of interest: end-to-end latency, packet loss rate, etc. Based on such metrics of network performance we would then wish to quantify the external loading point of a large system of interconnected queues at which the network changes its behaviour. For example, if packet loss rate is of interest, we wish to know the loading point at which this rate exceeds a given threshold. One of the important features of such a theory is that it must model all sources of stochasticity in the system of interconnected queues that form the core network. This implies that correlations between the states of queues, which have a knock-on effect on each other, must be modelled explicitly as interactions.

Constructing of this theory has attracted a lot of attention [4,5,6,7,8,9] but this work is far from over. In this paper we take a somewhat different approach

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 769–777, 2002.

in attempting to formulate a phenomenological theory of traffic in interacting queues. The chosen approach can be scaled to large numbers of queues which is essential in modelling a large-scale core IP network. The methodology adopted here is an adaptation of the methods of statistical mechanics, which are ideally suited to the study of large systems in the presence of sources of stochasticity.

The "microscopic" state variables which we choose to define queue dynamics, will here be the lengths of each and every buffer in the network. The sources of stochasticity we consider are the random packet lengths (measured in bits) and packet inter-arrival times. These latter random variables are distributed according to some probability distribution, which will not be discussed here (c.f. [1,2,3]).

2 Background

The average latency along some path, or the loss rate of a single buffer, or a group of buffers can be calculated using a joint probability distribution (PDF) function of all queue lengths of all routers at time t, $P = P(\{\ell_i\}_{i=1}^{N_\mathrm{q}}; t)$, where N_q is the number of queues in the network.

It is natural to expect that some queue lengths are highly correlated, whereas others are not. So, it is quite difficult to write a dynamical equation (e. g. a Fokker-Planck equation) for P and, never mind, solve it. For this reason we try to exploit some kind of approximation scheme and the simplest possible one is the mean-field approximation originating from Quantum Field Theory.

In the simplest case the mean-field approximation implies that the joint PDF can be represented as follows

$$P(\{\ell_i\}_{i=1}^{N_\mathrm{q}}; t) = \prod_{i=1}^{N_\mathrm{q}} p(\ell_i, t) \tag{1}$$

where $p(\ell_i, t)$ is the PDF of an individual queue and should be determined in a self-consistent manner to account for interactions. The dynamical equation for $p(\ell_i, t)$ of an individual queue should account for the fact that the queue interacts with an "average field" of all other queues, in this case an "average" network load. Individual PDF's in the right-hand side of (1) can actually depend on more than one variable ℓ for strongly correlated subsystems. This depends on the level of approximation we wish to use to account explicitly for certain types of correlation between queues. In this paper we consider two such subsystems: two cascaded queues and n queues attached to the same router.

It has been shown in [1,2,3] that the traffic in a core IP network acquires quasi-Markovian properties in and near its congested state. Hence, for a subsystem's PDF we can write down a Fokker-Planck equation [10]. Parameters entering the Fokker-Planck equation and, as a result, the PDF's themselves will depend on overall network load, capacity of lines and switches, average states of adjacent queues, etc., and should be determined in a self-consistent manner but this is beyond the scope of this paper.

3 Dynamical Model for Interacting Queues in Subsystem

As an example of a subsystem we first consider two cascaded queues consisting of an output buffer of a router connected to an input buffer of an adjacent router (see Fig. 1). Omitting the derivation details (which can be found in [10]), the Fokker-Planck (FP) equation for this system has the following form,

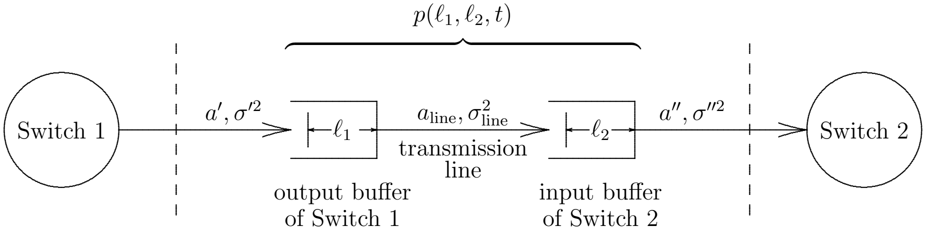


Fig. 1. Subsystem of two cascaded queues

$$\begin{aligned}\partial_t p(\ell_1, \ell_2, t) = &- \frac{\partial}{\partial \ell_1}\left[(a_1 - b_1\ell_1 - b_2\ell_2)p(\ell_1, \ell_2, t)\right] - a_2 \frac{\partial}{\partial \ell_2} p(\ell_1, \ell_2, t)\\ &+ \frac{\sigma_1^2}{2}\frac{\partial^2}{\partial \ell_1^2} p(\ell_1, \ell_2, t) + \frac{\sigma_2^2}{2}\frac{\partial^2}{\partial \ell_2^2} p(\ell_1, \ell_2, t) - \sigma^2_{\text{line}} \frac{\partial^2}{\partial \ell_1 \partial \ell_2} p(\ell_1, \ell_2, t)\end{aligned} \tag{2}$$

where

$$a_1 = \frac{a' - a_{\text{line}}}{\ell_{\max}}\ , \quad a_2 = \frac{a_{\text{line}} - a''}{\ell_{\max}}\ , \quad \sigma_1^2 = \frac{\sigma'^2 + \sigma^2_{\text{line}}}{\ell^2_{\max}}\ , \quad \sigma_2^2 = \frac{\sigma^2_{\text{line}} + \sigma''^2}{\ell^2_{\max}} \tag{3}$$

and natural boundary conditions are assumed [10]

$$\mathbf{n} \cdot \mathbf{J}(\ell_1, \ell_2, t)|_{(\ell_1, \ell_2) \in S} = 0 \tag{4}$$

Here we have introduced the concept of a probability current [10], which in our case is defined as follows

$$\mathbf{J} = \begin{pmatrix} J_1 \\ J_2 \end{pmatrix} \tag{5}$$

where

$$\begin{aligned} J_1 =& (a_1 - b_1\ell_1 - b_2\ell_2)p(\ell_1, \ell_2, t) - \frac{\sigma_1^2}{2}\frac{\partial}{\partial \ell_1} p(\ell_1, \ell_2, t) + \frac{\sigma^2_{\text{line}}}{2}\frac{\partial}{\partial \ell_2} p(\ell_1, \ell_2, t)\\ J_2 =& a_2 p(\ell_1, \ell_2, t) - \frac{\sigma_2^2}{2}\frac{\partial}{\partial \ell_2} p(\ell_1, \ell_2, t) + \frac{\sigma^2_{\text{line}}}{2}\frac{\partial}{\partial \ell_1} p(\ell_1, \ell_2, t)\end{aligned} \tag{6}$$

and $\mathbf{n}$ is a unit vector normal to the boundary S. The boundary S is a square with the sides at $\ell_1 = 0, \ell_1 = 1, \ell_2 = 0, \ell_2 = 1$. Note that we measure lengths of queues ℓ_1, ℓ_2 in terms of a fraction of the size of the corresponding buffer (here we set them all equal for the sake of simplicity), so that ℓ_1, ℓ_2 run from 0 to 1. Parameters $a', {\sigma'}^2$ characterise the mean value and the variance per unit time of the traffic coming into the first buffer [10]; $a_{\text{line}}, \sigma^2_{\text{line}}$ characterise the mean value and the variance of the traffic coming through the line; $a'', {\sigma''}^2$ characterise the mean value and the variance of the switching capacity available to the second buffer (traffic is measured in bits and all quantities are per unit time). We do not discuss here the nature of these parameters as they should be defined self-consistently with other parameters of a broader model for the overall network. Parameters b_1, b_2 are sensitivities of the routing protocol to the congestion of the queues. The dependence of the FP equation on the protocol sensitivities can be explained as follows. Relative queue lengths ℓ_1, ℓ_2 quantify the congestion level of the subsystem. The factor $a_1 - b_1\ell_1 - b_2\ell_2$ determines the average amount of traffic diverted to the subsystem: the more it is congested the less traffic (on average) is (or should be) diverted to it.

We seek an equilibrium (stationary) solution of the FP equation. For a equilibrium solution the detailed balance condition must be satisfied [10], and this condition demands that the parameters b_1, b_2 to be constrained as follows:

$$b_1 = b\cos\varphi \ , \quad b_2 = b\sin\varphi \ , \qquad \cos\varphi = \frac{\sigma_2^2}{\sqrt{\sigma_2^4 + \sigma_{\text{line}}^4}} \ , \quad \sin\varphi = \frac{\sigma_{\text{line}}^2}{\sqrt{\sigma_2^4 + \sigma_{\text{line}}^4}} \tag{7}$$

with b to be a single free parameter characterising the sensitivity of the protocol to the congestion of the subsystem as a whole. The stationary solution is

$$p^{(\text{s})}(\ell_1, \ell_2) = \mathcal{N}^{-1} \exp\left[-\frac{b}{\sigma^2}\left(\lambda_1 - \frac{\bar{a}}{b}\right)^2 + \frac{2a_2\lambda_2}{\sqrt{\sigma_2^4 + \sigma_{\text{line}}^4}} \right] \tag{8}$$

where

$$\lambda_1 = \ell_1\cos\varphi + \ell_2\sin\varphi \ , \qquad \lambda_2 = \ell_2\cos\varphi - \ell_1\sin\varphi \tag{9}$$

and

$$\sigma^2 = \frac{\sigma_1^2\sigma_2^2 - \sigma_{\text{line}}^4}{\sqrt{\sigma_2^4 + \sigma_{\text{line}}^4}} \ , \qquad \bar{a} = a_1 + a_2\frac{\sigma_{\text{line}}^2[\sigma_1^2 + \sigma_2^2]}{\sigma_2^4 + \sigma_{\text{line}}^4} \tag{10}$$

The normalisation constant $\mathcal{N}$ is a lengthy linear combination of error functions of different arguments, which is omitted here for the sake of brevity.

Using the stationary PDF and the conditional PDF $w(\ell_1', \ell_2', t'|\ell_1, \ell_2, t)$ for the transition from the state ℓ_1, ℓ_2 at time t to the state ℓ_1', ℓ_2' at time t' which is a solution of the same FP equation with the initial condition $w|_{t'=t} = \delta(\ell_1' - \ell_1)\delta(\ell_2' - \ell_2)$ we define the amount of the dropped traffic per unit time by the following expression (which should be considered as an estimation to the loss rate):

$$R_{\text{loss}} = \lim_{\Delta t \to 0} \frac{1}{\Delta t} \left(\int_1^\infty \mathrm{d}\ell_1' \int_{-\infty}^\infty \mathrm{d}\ell_2' \, (\ell_1' - 1) + \int_1^\infty \mathrm{d}\ell_2' \int_{-\infty}^\infty \mathrm{d}\ell_1' \, (\ell_2' - 1) \right) \\ \times \int_0^1 \mathrm{d}\ell_1 \int_0^1 \mathrm{d}\ell_2 \, w(\ell_1', \ell_2', t + \Delta t | \ell_1, \ell_2, t) p^{(\mathrm{s})}(\ell_1, \ell_2) \tag{11}$$

We only present here the final expression for the packet loss rate in the case when the capacity of the transmission line between the buffers is equal to the switching capacity available to the second buffer, $a'' = a_{\text{line}}$. In this case $a_2 = 0$ and the PDF (8) takes the form:

$$p^{(\mathrm{s})}_{a_2=0}(\ell_1, \ell_2) = \mathcal{N}^{-1}_{a_2=0} \exp\left[-\frac{b}{\sigma^2} \left(\lambda_1 - \frac{a_1}{b} \right)^2 \right] \tag{12}$$

The normalisation constant $\mathcal{N}_{a_2=0}$ is now determined by the following relation

$$\mathcal{N}_{a_2=0} = \frac{1}{\beta \sin 2\varphi} \Bigg(\mathrm{e}^{-\xi_{10}^2} - \mathrm{e}^{-\xi_{20}^2} - \mathrm{e}^{-\xi_{30}^2} + \mathrm{e}^{-\xi_{40}^2} \\ + \sqrt{\pi} \left[\xi_{10} \mathrm{erf}(\xi_{10}) - \xi_{20} \mathrm{erf}(\xi_{20}) - \xi_{30} \mathrm{erf}(\xi_{30}) + \xi_{40} \mathrm{erf}(\xi_{40}) \right] \Bigg) \tag{13}$$

where

$$\xi_{10} = -\frac{\alpha}{\sqrt{\beta}} \qquad \xi_{20} = \sqrt{\beta} \sin\varphi - \frac{\alpha}{\sqrt{\beta}} \qquad \xi_{30} = \sqrt{\beta} \cos\varphi - \frac{\alpha}{\sqrt{\beta}}$$

$$\xi_{40} = \sqrt{\beta}(\sin\varphi + \cos\varphi) - \frac{\alpha}{\sqrt{\beta}} \qquad \alpha = \frac{a_1}{\sigma^2} \qquad \beta = \frac{b}{\sigma^2}$$

For the loss rate we obtain

$$R^{a_2=0}_{\text{loss}} = \frac{1}{8\mathcal{N}_{a_2=0}} \sqrt{\frac{\pi}{\beta}} \left\{ \frac{\sigma_1^2}{\sin\varphi} \left[\mathrm{erf}(\xi_{40}) - \mathrm{erf}(\xi_{30}) \right] + \frac{\sigma_2^2}{\cos\varphi} \left[\mathrm{erf}(\xi_{40}) - \mathrm{erf}(\xi_{20}) \right] \right\} \tag{14}$$

The behaviour of the loss rate $R^{a_2=0}_{\text{loss}}$ is illustrated in Fig. 2.

It can be noticed that the loss rate is significantly greater than zero even below the naive capacity threshold $a' = 1$, especially in the presence of a routing protocol insensitive to congestion, or when the arriving traffic variance is relatively large.

A similar analysis for n interacting queues connected to the same switching device (see Fig. 3) has been completed. The Fokker-Planck equation for the PDF

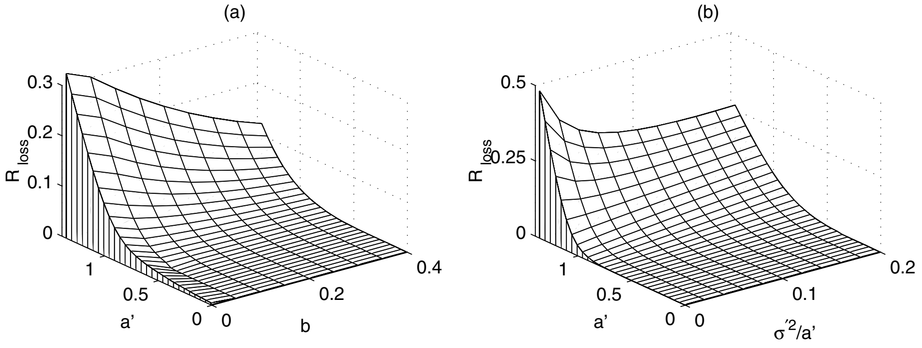


Fig. 2. Packet loss rate R_{loss} (in normalised units of buffer size $\ell_{\max}$ per unit time) plotted against traffic arrival rate a' (with a'' equal to the interconnecting line capacity $a_{\text{line}} = 1$). In subplot (a) this is also plotted against the protocol sensitivity b for a constant ratio of arriving traffic variance to the mean rate $\sigma'^2/a' = 0.1$, whereas in subplot (b) this is also plotted against the ratio σ'^2/a', for a constant protocol sensitivity $b = 0.1$. The theoretical capacity of this subsystem of queues is at $a'(= a'') = 1$.

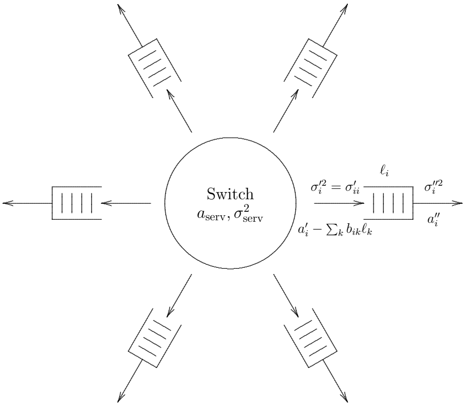

Switch
$a_{\text{serv}}, \sigma^2_{\text{serv}}$
ℓ_i
$\sigma_i'^2 = \sigma'_{ii}$
$\sigma_i''^2$
$a'_i - \sum_k b_{ik}\ell_k$
a''_i

Fig. 3. Subsystem of n queues connected to the same switching device.

of this subsystem, $p = p(\ell_1, \ldots, \ell_n; t)$, is

$$\partial_t p = -\sum_i \frac{\partial}{\partial x_i}\left[a_i - \sum_k b_{ik}\ell_k\right] p + \frac{1}{2}\sum_{i,k} \frac{\partial^2}{\partial x_i \partial x_k}\sigma_{ik} p \tag{15}$$

where

$$a_i = a'_i - a''_i \ , \qquad \sigma_{ik} = \sigma'_{ik} + \delta_{ik}\sigma_i''^2 \tag{16}$$

b_{ik} is the set of protocol sensitivities similar to the ones described earlier, a_{serv}, σ^2_{serv} is the mean value and variance (per unit time) of the overall traffic coming from the central switching device, $a'_i - \sum_k b_{ik}\ell_k$ is the set of mean values of traffic coming towards of individual queues, σ'_{ik} is the corresponding covariance matrix, a''_i, σ''^2_i the mean value and variance of switching capacities available at the other ends of individual queues. The equilibrium solution has the following form:

$$p^{(s)}(\ell_1, \ldots, \ell_n) = \mathcal{N}^{-1} \exp\left[2\sum_{ik}(\sigma^{-1})_{ik}\ell_i a_k - \sum_{ikj}(\sigma^{-1})_{ij} b_{jk}\ell_i\ell_k\right] \tag{17}$$

where $\mathcal{N}$ is the normalisation constant and the following set of relations (due to a detailed balance condition) should be imposed on the protocol sensitivities b_{ik}:

$$\sum_j b_{ij}\sigma_{jk} = \sum_j b_{kj}\sigma_{ji} \tag{18}$$

A general solution to (18) allows a large number of free parameters. In order to arrive at a reasonable number of those we impose the following restrictions:

$$a'_i = a_{\text{serv}} \frac{a''_i}{\sum_k a''_k}\ , \quad \sigma'_{ii} = \sigma'^2_i = \sigma'^2\ \forall i\ , \qquad \rho_{ik} \equiv \frac{\sigma'_{ik}}{\sigma'_i\sigma'_k} = \rho\ i \neq k \tag{19}$$

The first condition means that the mean value of traffic coming toward an individual queue is proportional to the switching capacity available to the corresponding queue (and vice versa), the second condition means that variances of traffic coming to all the queues are the same, and the third one is the statement that the correlation coefficients of any two pairs of incoming queue-traffic are equal as well, in order to maintain symmetry (this implicitly assumes that the system is homogeneous). In addition to this we are looking for a solution in the following class (the sensitivities of the idealised routing protocol for one particular queue to congestion on all other queues on the same switch are the same):

$$b_{ii} = \bar{b}_i\ \forall i\ , \qquad b_{ki} = \tilde{b}_i\ ,\ i \neq k \tag{20}$$

Then we have the following parameterisation for σ_{ik}, b_{ik}:

$$b_{ik} = \begin{cases} \frac{b}{\sigma^2_{\text{serv}}/n + \sigma''^2_i} & i = k \\ -\frac{1}{n-1}\frac{b}{\sigma^2_{\text{serv}}/n + \sigma''^2_i} & i \neq k \end{cases}\ , \quad \sigma_{ik} = \begin{cases} \sigma'^2 + \sigma''^2_i & i = k \\ \rho\sigma'^2 & i \neq k \end{cases} \tag{21}$$

where

$$\rho = -\frac{1}{n-1} + \frac{1}{n(n-1)}\frac{\sigma^2_{\text{serv}}}{\sigma'^2} \tag{22}$$

Summarising free parameters, we have: a_i'', $\sigma_i''^2$ are the mean value and variance per unit time of the switching capacities available to each queue, $a_{\mathrm{serv}}, \sigma_{\mathrm{serv}}^2$ are the mean value and variance per unit time of the overall traffic coming from the central switching fabric, σ'^2 (which is variance of the traffic coming to each queue) and b are actually characteristics of the routing protocol. The loss rate for this idealised router subsystem is defined an analogous fashion to (10) and has been computed explicitly. For reasons of economy of space, we omit the results from this paper, but will present them at the conference.

4 Preliminary Conclusions

We have presented a theoretical framework that can be used to model the interaction between a cascade of queues in a network. The theoretical model is capable of quantitative predictions of system throughput, loss rate, end-to-end delay, etc. Here we have presented the packet loss rate for a subsystem of two strongly correlated queues.

In the presence of stochastic packet lengths and inter-arrival times, we characterise the system in terms of mean arrival bit-rates identical to those of conventional queueing theory. However, here we also employ second-order statistics for the traffic, namely the variance of the arrival bit-rates. The latter parameters can be obtained either by observation on large-scale real networks, or as part of a broader model for an entire network.

As we can see from Fig. 2, the packet loss rate is non-zero for $a' < 1$ (i. e. arrival rates less than the system capacity) due to the presence of uncertainty in the arrival rate embodied in the variance terms. The packet loss rate can also be seen to reduce in the presence of a routing protocol that is sensitive to congestion. This type of analysis is clearly more useful than the conventional mean rate analysis (leaky bucket calculations) typically used in the first-order design of networks.

Finally, this methodology can be extended to a larger number of interacting queues in a straight-forward manner using functional integral methods. Such subsystems can be incorporated directly into broader network models (which need the various PDF's as input) in order to pursue the goal of arriving at a mathematical theory for dimensioning large-scale, core, packet-switched networks. Work on the detailed methodology behind such broader network models is at hand and will be presented in a series of papers in the future.

Acknowledgments. The authors would like to gratefully acknowledge the funding of this work by Nortel Networks. They also wish to acknowledge the helpful discussions they have had with Colin Camilleri of Nortel Networks which proved to be invaluable in distilling the operation of real network hardware into abstract mathematical models.

References

1. J. Cao, W. S. Cleveland, D. Lin, D. X. Sun, *The Effect of Statistical Multiplexing on the Long Range Dependence of Internet Packet Traffic*, Bell Labs Tech. Report (2002);
2. J. Cao, W. S. Cleveland, D. Lin, D. X. Sun, *Internet Traffic Tends To Poisson and Independent as the Load Increases*, http://citeseer.nj.nec.com/426819.html;
3. A. Stepanenko, C. C. Constantinou, T. N. Arvanitis and K. Baughan, *On the Statistical Properties of Core Network Internet Traffic*, submitted to Communications Letters (2001);
4. M. Schwartz: *Telecommunication Networks, Protocols, Modeling and Analysis*, Addison-Wesley (1987);
5. F. Kelly, *Loss networks,* Ann. Appl. Probab., **1**, pp. 319–378 (1991);
6. C. Graham and S. Méléard, *Chaos hypothesis for a system interacting through shared resources,* Probability Theory and Related Fields, **100**, pp.157–173 (1994);
7. N. D. Vvedenskaya, R. L. Dobrushin, and F. I. Karpelevich, *A queueing system with a choice of the shorter of two queues — an asymptotic approach,* Problems Inform. Transmission, **32**, pp. 15–27 (1996);
8. F. Delcoigne, G. Fayolle, *Thermodynamical limit and propagation of chaos in polling systems,* Markov Processes and Related Fields, **5**, pp. 89–124 (1999);
9. Guy Fayolle, Arnaud de La Fortelle, Jean-Marc Lasgouttes, Laurent Massouli'e, James Roberts, *Best-effort networks: modeling and performance analysis via large networks asymptotics*, IEEE INFOCOM 2001;
10. H. Risken, *The Fokker–Planck Equation: Methods of Solution and Applications*, 2$^{\mathrm{nd}}$ Edition, Springer Series in Synergetics, Springer-Verlag (1989);

Analysis of a MAC Protocol for a Time-Code Air Interface in LEO Mobile Satellite Systems

Romano Fantacci[1] and Giovanni Giambene[2]

[1] Dipartimento di Ingegneria Elettronica e Telecomunicazioni - Università degli Studi di Firenze, Via S. Marta, 3 - 50139 Firenze, ITALY,
fantacci@lenst.det.unifi.it,
[2] Dipartimento di Ingegneria dell'Informazione - Università degli Studi di Siena, Via Roma, 56 - 53100 Siena, ITALY,
giambene@unisi.it

Abstract. This paper deals with *Low Earth Orbit - Mobile Satellite Systems* (LEO-MSSs) and proposes a novel *Medium Access Control* (MAC) scheme for a hybrid time-code wideband air interface. This is a reservation scheme where each mobile terminal that needs to transmit makes random accesses (on available time-code resources) until it receives a positive acknowledgment from the satellite. Our protocol is named *Code Division-Packet Reservation Multiple Access scheme with Hindering States.* This work has been carried out within the "Multimedialità" Project of the Italian National Consortium for Telecommunications.

Index Terms : *Satellite Networks, Multiple Access Protocols, Quality of Service.*

1 Introduction

This paper proposes a modified version of the *Packet Reservation Multiple Access scheme with Hindering States* (PRMA-HS) [1] for the uplink of *Low Earth Orbit - Mobile Satellite Systems* (LEO-MSSs) with the SW-CTDMA (*Satellite Wideband - Code Time Division Multiple Access*) air interface defined in [2],[3]. This protocol has been named *Code Division-PRMA-HS* (CD-PRMA-HS). We assume that a *Mobile Terminal* (MT) must acquire the reservation of a slot-code per frame to transmit to the satellite. The near-far effect, typical of terrestrial cellular systems, is less evident in the satellite case, since all the users of a cell are about at the same distance from the satellite with about the same *Signal-to-Interference Ratio* (SIR). LEO satellites experience important distance variation due to their high-speed. We will refer to a Globalstar^{TM}-like LEO satellite constellation [4] with a minimum elevation angle of 15° (the maximum *Round Trip Delay*, RTD, value is 20 ms).

Among MTs, we have *Voice Terminals* (VTs) and *Data Terminals* (DTs) producing Web surfing traffic. Voice traffic belongs to the *conversational class*; a VT discards a packet from its buffer if it experiences a transmission delay

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 778–789, 2002.

greater than D_{max} (= 32 ms with standard voice codecs). The packet dropping probability, P_{drop}, must be lower than or equal to 1% (*Quality of Service*, QoS, requirement) for an acceptable voice quality [5]. Data traffic belongs to the *background class*. The QoS for data transmission is measured by the mean message delay (from the message generation to its complete transmission), T_{msg}.

2 System Description

We consider the quasi-synchronous return link envisaged in [3] for the *Frequency Division Duplexing* (FDD) version of the SW-CTDMA air interface [6]. Frames with length T_f (= 20 ms) are divided in N (= 8) slots of duration T_s. On a slot, N_{CODE} codes (= 8) are available for simultaneous transmissions. The elementary resource is a slot-code. We consider bursts (i.e., packets) of one slot [3],[6] and a fixed spreading level for VTs and DTs. With a QPSK modulation, a packet transports 640 bits with a payload L_p = 528 bits [3].

Each VT uses a slow speech activity detector to distinguish between *talkspurts* (ON state) and *silent pauses* (OFF state) within a conversation. In the ON state a VT generates one packet per frame [5]. As for a DT producing Web surfing traffic, we refer to the model and related parameter values shown in [7]: a DT alternates between a *datagram state* (DAT) where messages (i.e., datagrams) are generated and a *reading state* (READ) where no traffic is produced (see Fig. 1). The number of datagrams generated in the DAT state is geometrically distributed with expected value $m_{Nd} = 25$. The sojourn time in the READ state and the datagram interarrival time in the DAT state are exponentially distributed with mean values $m_{Dpc} = 4$ s and $m_{Dd} = 1/\ (2q)$ s, $q \in \{1, 2, 3, 4, 5, 6, 7\}$, respectively. The q value permits to modulate the DT traffic burstiness. The sojourn time in the DAT state is exponentially distributed, with mean value $m_{Lpc} = m_{Nd}\ m_{Dd}$. Thus, it is easy to show that the DT traffic source is a *2-state Markov Modulated Poisson Process* (2-MMPP).

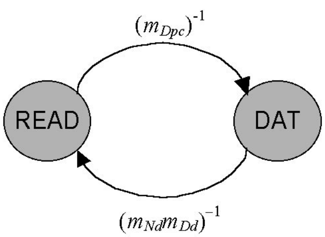


Fig. 1. Adopted model for the DT traffic source.

The mean datagram arrival rate is $\lambda_d = m_{Lpc}/\{m_{Dd}(m_{Lpc}+m_{Dpc})\}$. Datagrams have a random length in bytes according to the truncated Pareto probability density function shown in [7]. Datagrams are fragmented in packets before transmission. According to the packet payload, we have obtained the mean (mean square) datagram length as $L_d \approx 6$ packets/datagram ($L_{dq} \approx 830$ packets^2/datagram).

We require $T_f \geq RTD_{max}$, the maximum RTD value experienced by an MT within a cell, so that an MT attempting a transmission on a slot-code receives the outcome before the beginning of the same slot in the next frame. For a conservative study, we consider RTD $\equiv RTD_{max}$ and $T_f \approx n\ RTD_{max}$, where n is a divisor of N; the slot duration is $T_s = T_f/N$.

We assume that M_v VTs and M_d DTs share the use of the same carrier. VTs (DTs) acquire reservations on a talkspurt (datagram) basis. As soon as an MT receives the first packet in its idle buffer, it enters the contending state to acquire a reservation: the MT transmits this packet on available slot-codes, according to its permission probability. Transmission attempts are random in the time-code domain. *Orthogonal Variable Spreading Factor* (OVSF) codes are used by an MT for making simultaneous attempts (i.e., on the same slot) on different codes. The permission probabilities have been denoted by p_v and p_d for VTs and DTs, respectively. We have considered $p_v > p_d$, since VTs have a higher service priority than DTs (see Sections 5 and 7). Simultaneous attempts of different MTs on the same code collide and no reservation is achieved. The satellite sends a reservation notification to the MT, when it successfully decodes its first packet on a slot-code. While waiting for this message, the MT may undertake new random attempts on idle slots. Thus, if the previous attempt has been unsuccessful, this strategy allows a fast re-attempt scheme; otherwise, these further attempts are discarded by the satellite and may hinder the accesses of other MTs. The MT releases the reservation by setting an *end-of-transmission flag* in the header of the last packet. If a datagram arrives when a DT has already an active transmission with the satellite, the DT maintains the reservation until its buffer is emptied (exhaustive discipline).

OVSF codes are necessary when a multimedia terminal must simultaneously transmit VT and DT traffics. Joint detection is used in uplink, so that the intra-cell interference has a negligible impact on SIR. Thus, we assume that the system can support N_{CODE} different MTs transmitting on a given slot with a sufficient quality (outage events are not addressed here).

3 System Model

VT and DT behaviors are modeled by two discrete-time Makov chains, where state transitions occur at the slot end. The DT state diagram is shown in Fig. 2 (symbols A_d, U_d, σ_d, σ_{dp} and γ_{fd} of Fig. 2 are defined later). The VT state diagram is characterized by the states enclosed by a box in Fig. 2, where we consider the following probabilities [1]: $\gamma_v = 1 - e^{-T_s/t_1}$, $\sigma_v = 1 - e^{-T_s/t_2}$, $\gamma_{fv} = 1 - (1 - \gamma_v)^N$; finally, the definitions of A_v and U_v are given below. Starting from the silent pause (SIL), the VT enters the contending (CON) state when a new talkspurt is generated. If the VT makes a successful attempt, the VT enters the block of hindering states from HIN(N - 1) to HIN(N - N/n). In the HIN(N - N/n) state the VT receives the positive acknowledgment from the satellite; then,

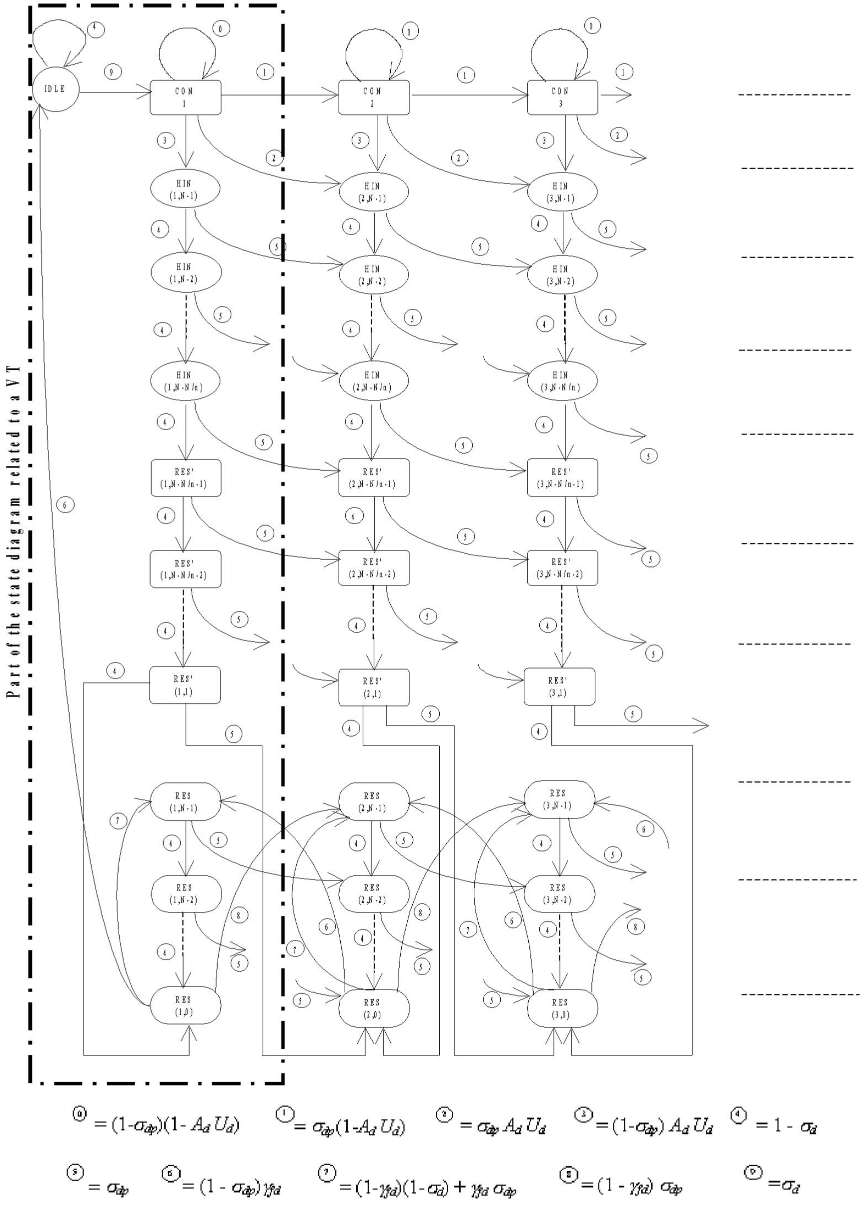


Fig. 2. DT and VT state diagrams.

the VT enters the block from RES'(N - N/n - 1) to RES'(1) due to the waiting time before transmitting the next packet on the reserved slot-code. The chain of a DT (Fig. 2) has an infinite number of states (each column of states is for a different number of messages in the DT buffer). The CON(j) state contains all the DTs that need to acquire a reservation with j messages in the buffer. Whereas, HIN(j , i), RES'(j , i) and RES(j , i) states are related to the slot-code reserved by the DTs with j datagrams in their buffers.

The system is characterized by the aggregated state { C_v, C_d, H_v, H_d, R_v^*, R_d^* }, where C_v (C_d) = number of VTs (DTs) in the CON state, H_v (H_d) = number of VTs (DTs) in the HIN states, R_v^* (R_d^*) = number of VTs (DTs) that know to have a reservation. Referring to a particular state, we define:

- The probability that a slot-code is not reserved, P_f:

$$P_f = 1 - \left(\frac{R_v^* + R_d^* + H_v + H_d}{NN_{CODE}} \right) \quad . \tag{1}$$

- The probability that a VT (DT) makes a transmission attempt on a slot-code, A_v (A_d):

$$A_v = p_v P_f \ \text{ and } \ A_d = p_d P_f \quad . \tag{2}$$

- The probability of a successful transmission attempt for a VT, U_v:

$$U_v = \begin{cases} (1-p_d)^{C_d+H_d} (1-p_v)^{C_v+H_v-1}, & \text{if } C_v \geq 1, \ \forall \ C_d, H_v, H_d \\ 0, & \text{if } C_v = 0, \ \forall \ C_d, H_v, H_d \end{cases} \tag{3}$$

  The corresponding expression for a DT, U_d, is obtained by changing subscript v with d and vice versa in (3).

A DT in the IDLE state enters the CON(1) state with probability σ_d, i.e., the probability that the sojourn time in the READ state ends within T_s; we have:

$$\sigma_d = 1 - e^{-\frac{T_s}{m_{D_{pc}}}} \quad . \tag{4}$$

Transitions between parallel columns in the diagram in Fig. 2 are made according to probability σ_{dp} (a DT can receive datagrams at any instant in the DAT state):

$$\sigma_{dp} = \left(1 - \frac{1}{N_{Dd}} \right) \left(1 - e^{-\frac{T_s}{m_{D_d}}} \right) \quad . \tag{5}$$

Finally, symbol γ_{fd} in Fig. 2 represents the probability that the sojourn time in the DAT state ends in the current frame. In particular, we have: γ_{fd} = Prob.{last datagram of the DAT state} × Prob.{last packet of the datagram}:

$$\gamma_{fd} = \frac{1}{N_{Dd}} \left[\frac{k}{(L_{d\_max} - 1) L_p + 1} \right]^{\upsilon} \tag{6}$$

where, according to [7], $k = 81.5$, $\upsilon = 1.1$ and $L_{d\_max} = \lceil 66666/L_p \rceil$, being $\lceil . \rceil$ the *ceiling function*.

Depending on the utilization of slot-codes by VTs and DTs, η_v and η_d, we obtain the total throughput η_{tot} and the related stability condition as:

$$\eta_{tot} = \eta_v + \eta_d = \frac{\psi_v M_v \left(1 - P_{drop}\right)}{NN_{CODE}} + \frac{\lambda_d T_f L_d M_d}{NN_{CODE}} < 1 \quad \left[\frac{packets}{slot\ code}\right] \quad . \tag{7}$$

where ψ_v denotes the voice activity factor and the packet dropping probability P_{drop} will be evaluated in Section 6.

Since a DT with a reservation uses one slot-code per frame, the following DT buffer stability condition must be fulfilled by the DT traffic load, ρ_d:

$$\rho_d = \lambda_d T_s N L_d < 1 \quad \left[\frac{packets}{DT\ frame}\right] \quad . \tag{8}$$

A new Web session is accepted if conditions (7) and (8) are fulfilled (connection admission control).

4 Equilibrium Point Analysis

The standard methods for discrete-time Markov-chains cannot be adopted here, since the number of states of the system exponentially increases with the number of MTs. Hence, we have used the *Equilibrium Point Analysis* (EPA) [5]; we study the equilibrium variables (denoted by small letters) by assuming that all the codes of a slot have the same probability to be utilized. Similarly to [1], we obtain the following system in the four unknown terms c_v, c_d, h_v, h_d:

$$M_v = \left[1 + \frac{\gamma_v}{\sigma_v}\right] c_v + \left[\frac{1}{\sigma_v} + \frac{N}{\gamma_{fv}}\right] N_{CODE} h_v \tag{9}$$

$$M_d = \frac{c_d + \frac{N_{CODE}}{\sigma_d} h_d}{1 - \lambda_d T_s L_d N} \tag{10}$$

$$h_v - u_v \left(1 - \gamma_v\right) p_v c_v \left[1 - \frac{h_v}{\gamma_{fv}} - \frac{\lambda_d T_s L_d \left(c_d + \frac{N_{CODE}}{\sigma_d} h_d\right)}{N_{CODE}\left(1 - \lambda_d T_s L_d N\right)}\right] = 0 \tag{11}$$

$$h_d - u_d p_d c_d \left[1 - \frac{h_v}{\gamma_{fv}} - \frac{\lambda_d T_s L_d \left(c_d + \frac{N_{CODE}}{\sigma_d} h_d\right)}{N_{CODE}\left(1 - \lambda_d T_s L_d N\right)}\right] = 0 \tag{12}$$

where c_v (c_d) is the equilibrium number of VTs (DTs) in the contending state(s), h_v (h_d) is the equilibrium number of VTs (DTs) in each hindering state per code, u_v is defined as:

$$u_v = \begin{cases} (1 - p_d)^{c_d + \frac{NN_{CODE}}{n} h_d} (1 - p_v)^{c_v + \frac{NN_{CODE}}{n} h_v - 1}, c_v \geq 1, \ \forall c_d, h_v, h_d \\ (1 - p_d)^{c_d + \frac{NN_{CODE}}{n} h_d} (1 - p_v)^{\frac{NN_{CODE}}{n} h_v}, 0 \leq c_v \leq 1, \quad \forall c_d, h_v, h_d \\ 0 \ \ , \ \ c_v = 0, \forall c_d, h_v, h_d \end{cases} \tag{13}$$

and u_d is obtained by changing subscript v with d (and vice versa) in u_v.

Since u_v and u_d have transcendent expressions, we numerically solve the EPA system (9)-(12) by using a four-dimensional recursive approach (the obtained solution depends on the starting point, if there are multiple EPA solutions).

5 Admissible Range for Permission Probabilities

We need to identify the values of p_v and p_d that allow a single and stable EPA solution. This problem can be solved by adopting the *catastrophe theory* that is based on a potential function V defined in the *state space* $\Omega_s = \{(c_v, c_d) \in \Re \times \Re$: $0 \leq c_v \leq M_v$ and $0 \leq c_d \leq M_d$ and depending on the *control space* $\Omega_c = \{(p_v, p_d) \in \Re \times \Re$: $0 < p_v$, $p_d < 1\}$ $V \; : \; \Omega_s \times \Omega_c \to \Re, \;\; V = V\left(\{c_v, c_d\}, \{p_v, p_d\}\right)$. EPA equations (9) and (10) (where $h_v = h_v(\{c_v, c_d\}$, $\{p_v, p_d\})$ and $h_d = h_d(\{c_v, c_d\}$, $\{p_v, p_d\})$ are assumed to be implicitly expressed in terms of c_v and c_d by solving (11) and (12)) are equivalent to the null gradient condition for V with respect to Ω_s, $-\nabla_s V = 0$:

$$\begin{cases} \frac{\partial V}{\partial c_v} = \left[1 + \frac{\gamma_v}{\sigma_v}\right] c_v + \left[\frac{1}{\sigma_v} + \frac{N}{\gamma_{fv}}\right] N_{CODE} h_v\left(\{c_v, c_d\}, \{p_v, p_d\}\right) - M_v = 0 \\ \frac{\partial V}{\partial c_d} = \frac{c_d + \frac{N_{CODE}}{\sigma_d} h_d(\{c_v, c_d\}, \{p_v, p_d\})}{1 - \lambda_d T_s L_d N} - M_d = 0 \end{cases} . \tag{14}$$

The EPA solutions are the *critical points* of V, $\zeta \in \Omega_s \times \Omega_c : -\nabla_s V|_\zeta = 0$. In [5], for a similar problem, the author proposes the simplifying assumption $c_d = M_d$. This approach does not allow a correct identification of the region in Ω_c with multiple EPA solutions (i.e., *bistable protocol behavior*), since the substitution $c_d = M_d$ strongly modifies the characteristics of the EPA system. Hence, the critical points can be studied by means of the *Hessian matrix*, $H_s(V)$ [8]:

$$H_s(V) = \begin{pmatrix} \frac{\partial^2 V}{\partial {c_v}^2} & \frac{\partial^2 V}{\partial c_d \partial c_v} \\ \frac{\partial^2 V}{\partial c_v \partial c_d} & \frac{\partial^2 V}{\partial {c_d}^2} \end{pmatrix} . \tag{15}$$

The signs of the eigenvalues φ_1 and φ_2 of the Hessian matrix in a critical point ζ identify if ζ represents a *minimum* (two positive eigenvalues ⇒ stable equilibrium point), a *maximum* (two negative eigenvalues) or a *saddle* point (two eigenvalues with different signs). Of course, the correct protocol behavior is allowed only in the region in Ω_c where there is a single solution of (14) that corresponds to a minimum of V. If we vary p_v and p_d in Ω_c, V modifies its shape and, correspondingly, the number of critical points (EPA solutions). Shape changes occur when the determinant of the Hessian matrix vanishes in critical points:

$$-\nabla_s V|_\zeta = 0 \;\; \text{and} \;\; \det\left\{ H_s(V)|_\zeta \right\} = 0 \quad . \tag{16}$$

For a given $(p_v$, $p_d) \in \Omega_c$ we need to evaluate $\det\{H_s(V)\}$ in each EPA solution to identify where condition (16) is fulfilled (= *bifurcation set* [8]). Unfortunately, this method cannot be practically implemented, since we do not (*a priori*) know how many solutions the EPA system admits for each $(p_v$, $p_d)$. This is the reason why we propose the following heuristic approach. By means of an EPA graphical solution, we have noticed that: (a) the EPA system always admits a solution close to the origin; (b) multiple EPA solutions always entail three EPA solutions; in such cases, the solution with the highest c_v and c_d values (*degenerate solution*) is characterized as:

$$c_v \approx c_{v,deg} = \frac{M_v}{\left[1 + \frac{\gamma_v}{\sigma_v}\right]} \quad , \quad c_d \approx c_{d,deg} = M_d \left(1 - \lambda_d T_s L_d N\right) \quad . \tag{17}$$

Practically, (17) is an EPA solution when, according to (9) and (10) $h_v \approx 0$ and $h_d \approx 0$. Hence, we use a four-dimensional iterative approach to solve the EPA system for given M_v and M_d values in two cases: (i) starting point $c_v = 0$, $c_d = 0$, $h_v = 0$, $h_d = 0$ so as to find the solution closer to the origin; (ii) starting point $c_v = c_{v,deg}$, $c_d = c_{d,deg}$, considering of course $h_v = h_d = 0$. There are three EPA solutions only when the EPA system admits different solutions in cases (i) and (ii). Accordingly, we scan Ω_c to identify the regions where there are one or three EPA solutions. Thus, in these different cases we may verify (16) to identify the bifurcation set (see below the derivation of the components of the Hessian matrix). Fig. 3 shows the bifurcation set in the control space Ω_c with $q = 1$, $n = 1$ for $M_v = 64$ VTs, $M_d = 40$ DTs. We note that the region with a single EPA solution obtained here is different from the region identified according to the method proposed in [5].

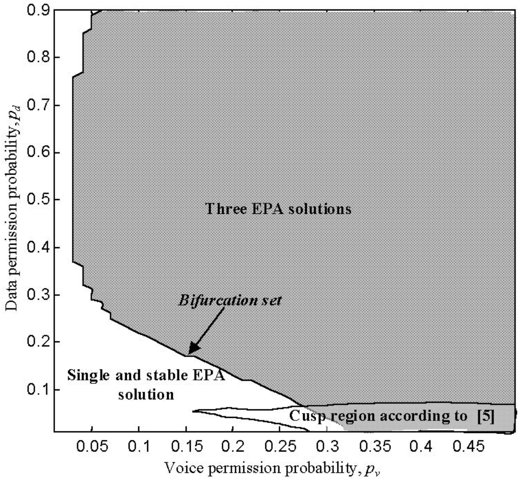


Fig. 3. Bifurcation set in the control space Ω_c for $M_v = 64$ VTs, $M_d = 40$ DTs, $q = 1$ and $n = 1$. This figure also contains the cusp region according to [5].

We can characterize the system behavior corresponding to the critical points of the potential function by evaluating the eigenvalues of the Hessian matrix (15), whose entries are obtained as follows:

$$\begin{array}{l}
\frac{\partial^2 V}{\partial {c_v}^2} = 1 + \frac{\gamma_v}{\sigma_v} + \left[\frac{1}{\sigma_v} + \frac{N}{\gamma_{fv}}\right] N_{CODE} \frac{\partial h_v(\{c_v, c_d\}, \{p_v, p_d\})}{\partial c_v} \\
\frac{\partial^2 V}{\partial {c_d}^2} = \frac{1 + \frac{N_{CODE}}{\sigma_d} \frac{\partial h_d(\{c_v, c_d\}, \{p_v, p_d\})}{\partial c_d}}{1 - \lambda_d T_s L_d N} \\
\frac{\partial^2 V}{\partial c_d \partial c_v} = \left[\frac{1}{\sigma_v} + \frac{N}{\gamma_{fv}}\right] N_{CODE} \frac{\partial h_v(\{c_v, c_d\}, \{p_v, p_d\})}{\partial c_d} \\
\frac{\partial^2 V}{\partial c_d \partial c_v} = \frac{\partial^2 V}{\partial c_v \partial c_d} \;\; (Hessian\ matrix\ symmetry\ condition) \Rightarrow \\
\Rightarrow \frac{\partial h_d(\{c_v, c_d\}, \{p_v, p_d\})}{\partial c_v} = \sigma_d \left[\frac{1}{\sigma_v} + \frac{N}{\gamma_{fv}}\right] \left[1 - \lambda_d T_s L_d N\right] \frac{\partial h_v(\{c_v, c_d\}, \{p_v, p_d\})}{\partial c_d}
\end{array} \quad (18)$$

We obtain $\frac{\partial h_v}{\partial c_v}$, $\frac{\partial h_v}{\partial c_d}$, $\frac{\partial h_d}{\partial c_v}$, and $\frac{\partial h_d}{\partial c_d}$ by using the theorem for the derivative of the implicit functions applied to (11) and (12) and by using the symmetry condition given by the last equation in (18). Hence, for each EPA solution (c_v, c_d), we solve a system with four transcendent equations in the unknown $\frac{\partial h_v}{\partial c_v}$, $\frac{\partial h_v}{\partial c_d}$, $\frac{\partial h_d}{\partial c_v}$, and $\frac{\partial h_d}{\partial c_d}$ by means an iterative method with starting point (0, 0, 0, 0). Accordingly, we have verified that the EPA solution is always stable in the region in Ω_c where there is a single EPA solution; p_v and p_d values will be selected in this region, as explained in Section 7.

6 Performance Analysis

In order to obtain P_{drop} we condition on the aggregated state { C_v, C_d, H_v, H_d, R_v^*, R_d^* }. The joint state probability distribution can be derived according to [1] and by considering that there are now NN_{CODE} resources. P_{drop} is obtained on the basis of the method shown in [1], where we have to use a different expression of the conditioned probability that a VT in the CON state has at least one successful attempt on a slot, $P_{s,v}(C_v, C_d, H_v, H_d, R_v^*, R_d^*)$:

$$P_{s,v}\left(C_v, C_d, H_v, H_d, R_v^*, R_d^*\right) = 1 - \left(1 - A_v U_v\right)^{N_{CODE}} \quad . \quad (19)$$

The main original aspect for the performance analysis with respect to [1] is given by the derivation of T_{msg}. In particular, T_{msg} is the sum of two contributions: (*i*) the mean access delay, T_{acc}, that is the mean time needed by the DT to acquire a reservation (only for a datagram arrived at an empty DT buffer); (*ii*) the mean transmission delay T_{delay} due to a queuing system of the *2-MMPP*$^{[P]}$*/D/1* type (where "*2-MMPP*$^{[P]}$" stands for the 2-MMPP arrival process of datagrams of a DT, "*D*" stands for a deterministic packet transmission time of a frame, "1" means that only one packet can be transmitted per slot).

$$T_{msg} = T_{acc} + T_{delay} \quad . \quad (20)$$

T_{acc} can be approximated[1] as the product of the mean time a DT spends in the contending phase, $E[t_{CON_d}]$, and the probability that an arriving datagram finds the DT in the IDLE state, P_{idle}:

[1] We have considered here that T_{acc} is a pure delay contribution; any queuing phenomenon is negligible in the short time spent by a DT in CON states.

$$T_{acc} = E\left[t_{CON_d}\right] P_{idle} \tag{21}$$

According to the EPA system, $P_{idle} = s_d/M_d = N_{CODE}h_d/(\sigma_d M_d)$. Moreover, $E[t_{CON_d}]$ can be derived by using the same approach proposed in [1] considering that there are NN_{CODE} resources. We focus now on T_{delay}. We start by deriving the mean packet delay, T_{pkt}. We embed the model to the end of slots and we modify the approximated analytical approach proposed in [9] by taking into account that packets have a compound arrival process due to both the datagram generation process and the variable length of each arrival. The packet arrival process is characterized by the following probability-generating matrix:

$$\mathbf{Q}\left(z\right) = \begin{bmatrix} p_{11}e^{\lambda_p T_s [L(z)-1]} & p_{12} \\ p_{21}e^{\lambda_p T_s [L(z)-1]} & p_{22} \end{bmatrix} \tag{22}$$

where $\lambda_p = 2q$ is the mean datagram arrival rate in the DAT state; $L(z)$ is the probability-generating function of the datagram length distribution in packets; $p_{12} = 1 - e^{-T_s/m_{Lpc}}$ is the probability that the source leaves the DAT state in T_s; p_{11} = 1 - p_{12}; $p_{21} = 1 - e^{-T_s/m_{Dpc}}$ is the probability that the source leaves the READ state in T_s; p_{22} = 1 - p_{21}.

Let $\mathbf{s} = (s_1\ ,\ s_2)^T$, where (apex T denotes transposition) s_1 is the probability of the DAT state and s_2 is the probability of the READ state.

$$s_1 = \frac{p_{21}}{p_{12}+p_{21}} \quad , \quad s_2 = \frac{p_{12}}{p_{12}+p_{21}} \quad . \tag{23}$$

Since each source with a reservation transmits one packet per frame, the service time of a packet is equal to N slots. According to [9] and on the basis of (22), the mean packet delay, T_{pkt}, results as:

$$T_{pkt} = N + \frac{\lambda_1^{''}\left(1\right)N^2}{2\rho_d\left[1-\rho_d\right]} + \frac{N\xi_1^{'}\left(1\right)}{\rho_d} \quad [slots] \tag{24}$$

where $\lambda_1^{''}(1)$ and $\xi_1^{'}(1)$ have quite complex definitions [9]:

$$\lambda_1^{'}\left(1\right) = \lambda_d T_s L_d \quad \text{and} \quad \lambda_2^{'}\left(1\right) = p_{11}\lambda_p T_s L_d - \lambda_1^{'}\left(1\right) \tag{25}$$

$$\lambda_1^{''}\left(1\right) = \lambda_1^{'}\left(1\right)\left[\frac{L_{dq}-L_d}{L_d} + \lambda_p T_s L_d + 2\frac{\lambda_2^{'}\left(1\right)}{p_{12}+p_{21}}\right] \tag{26}$$

$$\begin{array}{l} \xi_1^{'}\left(1\right) \cong \left\{\mathbf{s}^T\mathbf{Q}\left(0\right)\mathbf{u}_1^{'}\left(1\right)\right\} / \left\{\mathbf{s}^T\mathbf{Q}\left(0\right)\mathbf{1}\right\} \\ \text{being} \quad \mathbf{u}_1^{'}\left(1\right) = \frac{\lambda_2^{'}(1)}{p_{12}}\left(s_2, -s_1\right)^T \quad \text{and} \quad \mathbf{1} = (1,1)^T \end{array} \quad . \tag{27}$$

Since T_{pkt} is related to the transmission of a packet in the middle of a datagram, we may consider that T_{delay} is obtained by adding to T_{pkt} the transmission of the remaining half datagram:

$$T_{delay} = T_{pkt} + \frac{L_d}{2}N \quad [slots] \quad . \tag{28}$$

7 Results

A CD-PRMA-HS simulator has been realized and very long simulations (about 200×10^6 slots) have been performed to achieve very accurate results. We have carried out a first group of simulation runs for different values of p_v and p_d in Ω_c in a typical configuration with $M_v = 64$ VTs, $M_d = 44$ DTs, $q = 1$ and $n = 1$ (these values allow the fulfillment of conditions (7) and (8)). We have obtained that P_{drop} has a good behavior for p_v around 0.15 - 0.2 and P_{drop} mildly increases with p_d. Thus, we have selected $p_v = 0.2$ and $p_d = 0.1$. On the basis of Fig. 3, such choice allows a single and stable EPA solution for the typical M_v, M_d, q and n values considered in this paper. Figs. 4 and 5 compare analytical and simulation results as a function of q for $p_v = 0.2$, $p_d = 0.1$, $n = 1$ and considering two cases: (i) $M_v = 64$ VTs and $M_d = 44$ DTs; (ii) $M_v = 95$ VTs and $M_d = 9$ DTs.

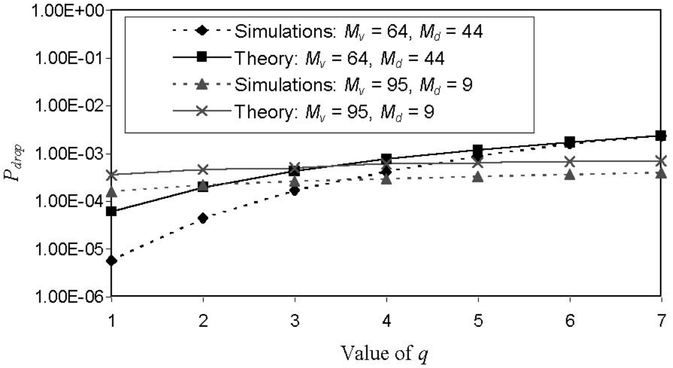


Fig. 4. Comparison of P_{drop} obtained from theory and simulations.

We note that the P_{drop} theory gives a sufficiently close upper bound to simulation results. Moreover, in case (i) the presence of a higher number of DTs entails that the increase of q has a more significant impact on P_{drop} than in case (ii). As for T_{msg}, the proposed 2-MMPP analytical approach gives a conservative estimate of the CD-PRMA-HS performance that permits to characterize the service experienced by Web traffic sources. Of course, T_{msg} increases with q. Moreover, T_{msg} results do not appreciably differ in the two cases, since T_{msg} is dominated by T_{delay} that is about the same in the two cases. Let us refer to the maximum number of VTs, M_{vmax}, that permits to have $P_{drop} \leq 1\%$, a single and stable solution with $p_v = 0.2$ and $p_d = 0.1$; correspondingly, we can consider the total throughput η_{tot}. Even considering the very bursty DT traffic case with $q = 5$, we have that M_{vmax} is around to 67 VTs for $M_d = 55$ DTs with $n = 1$, thus achieving a high η_{tot} value about equal to 0.82 packets/slot-code.

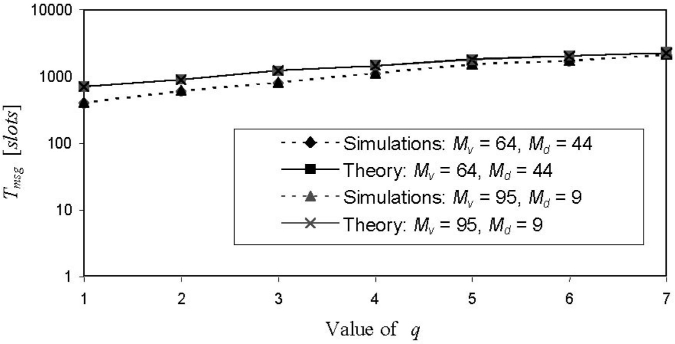


Fig. 5. Comparison of T_{msg} obtained from theory and simulations.

8 Conclusions

The new CD-PRMA-HS protocol has been proposed in this paper for the SW-CTDMA air interface in LEO-MSSs, supporting both conversational and background traffics. A stability study and a performance analysis have permitted to characterize the behavior of this new protocol.

References

1. Benelli, G., Fantacci, R., Giambene, G., Ortolani, C.: Voice and Data Transmissions with a PRMA-like Protocol in High Propagation Delay Cellular Systems. IEEE Trans. on Veh. Tech. **49** (2000) 2126–2147.
2. Taaghol, P., Evans, B. G., Buracchini, E., De Gaudenzi, R., Gallinaro, G., Ho Lee, J., Gu Kang, C.: Satellite UMTS/IMT2000 W-CDMA Air Interfaces. IEEE Comm. Mag. **37** (1999) 116–126.
3. ESA. Wideband Hybrid CDMA/TDMA Option for the Satellite Component of IMT-2000 "SW-CTDMA". ESA Proposal of a Candidate RTT, V1.0, 29 June 1998.
4. Official web site with URL: http://www.globalstar.com.
5. Nanda, S.: Stability Evaluation and Design of the PRMA Joint Voice Data System. IEEE Trans. on Comm. **42** (1994) 2092–2104.
6. ESA. Evaluation Report by European Space Agency IMT-2000 Satellite RTT Evaluation Committee. Sept. 30, 1998.
7. ETSI. Selection Procedures for the Choice of Radio Transmission Technologies of the UMTS (UMTS 30.03 Version 3.1.0). ETSI, Nov. 1997.
8. Saunders. An Introduction to Catastrophe Theory. Cambridge University press, NY, USA, 1980.
9. Steyaert, B., Bruneel, H., Petit, G. H., De Vleeschauwer, D.: A Versatile Queueing Model Applicable in IP Traffic Studies. COST 257 Project, TD (00)-02, Jan. 2000.

Performance Analysis of LEO Satellite Networks

A. Halim Zaim, Harry G. Perros, and George N. Rouskas

Department of Computer Science, North Carolina State University, Raleigh, NC, USA
ahzaim,hp,rouskas@eos.ncsu.edu

Abstract. We present an analytical model for computing call blocking probabilities in a LEO satellite network that carries voice calls. Both satellite-fixed and earth-fixed constellations with inter-orbit links and hand-offs are considered. The model is analyzed approximately by decomposing it into sub-systems. Each sub-system is solved in isolation exactly using a Markov process, and the individual results are combined together through an iterative method. Numerical results demonstrate that our method is accurate for a wide range of traffic patterns.

1 Introduction

Recent advances in satellite communications make it possible to use satellites as an alternative to wireless telephone and wireless networks. A low (or medium) earth orbit (LEO or MEO) satellite system is a set of identical satellites, launched in several orbital planes with the orbits having the same altitude. The satellites move in a synchronized manner in trajectories relative to the earth. Such a set of satellites is referred to as a *constellation* of satellites.

If satellites are equipped with advanced on-board processing, they can communicate directly with each other by line of sight using inter-satellite links (ISL). If the ISL is between satellites on the same orbit, it is called intra-plane ISL, and if it is between satellites in adjacent planes it is called inter-plane ISL. Depending on the antenna technology used, satellite constellations can provide one of two types of coverage. If the satellite antenna is fixed as the satellite moves along its orbit, then the coverage is called *satellite-fixed*. In this case, the footprint area moves along with the satellite. In *earth-fixed coverage*, the earth's surface is divided into cells, as in a terrestrial cellular system, and a cell is serviced continuously by the same beam during the entire time that the cell is within the footprint area of the satellite. This type of coverage requires an antenna that tracks the cell area.

The performance of satellite systems has been studied by several authors [1]-[8]. In general, most studies rely on simple queueing models (e.g., the M/M/K/K queue, where K denotes the number of channels per cell) to evaluate call blocking probabilities, and focus on devising methods for improving the performance of calls during hand-offs (e.g., by assigning higher priority to hand-off calls, using guard channels, or making reservations ahead of a hand-off instant). In [9], the authors proposed an approximation method for calculating call blocking probabilities in a group of LEO/MEO satellites arranged in a single orbit. In this

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 790–801, 2002.

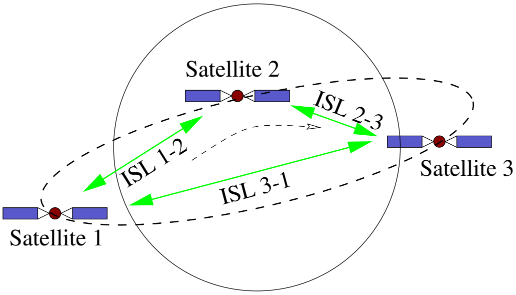


Fig. 1. Three satellites in a single orbit

paper we generalize this algorithm to an entire constellation of LEO/MEO satellites involving multiple orbits. We consider both satellite-fixed and earth-fixed constellations with inter-orbit links and hand-offs.

The paper is organized as follows. In Section 2 we present briefly an exact Markov process model under the assumption that satellites are fixed in the sky (i.e., no hand-offs take place), and in Section 3 we present an approximate decomposition algorithm for a constellation of satellites. In Section 4 we extend our approach to model hand-offs for both earth-fixed and satellite-fixed coverage. We present numerical results in Section 5, and in Section 6 we conclude the paper.

2 An Exact Model for the No Hand-Offs Case

In this section we review briefly the single-orbit model proposed in [9]. This model is used in the decomposition algorithm described in the following section.

Let us consider a single orbit of a constellation, and let us assume that the position of the satellites is fixed in the sky, as in the case of geostationary satellites. The analysis of such a system is simpler, since no calls are lost due to hand-offs from one satellite to another, as when the satellites move with respect to the users on the earth. This model is used in Section 4 to model both earth-fixed and satellite-fixed systems with hand-offs.

Each up-and-down link (UDL) of a satellite has capacity to support up to C_{UDL} bidirectional calls, while each inter-satellite link (ISL) has capacity equal to C_{ISL} bidirectional calls. We assume that call requests arrive at each satellite according to a Poisson process, and that call holding times are exponentially distributed. We now show how to compute blocking probabilities for the 3 satellites in the single orbit of Figure 1. The analysis can be generalized to analyze $k > 3$ satellites in a single orbit. For simplicity, we consider only shortest-path routing, although the analysis can be applied to any fixed routing scheme.

Let n_{ij} be a random variable representing the number of active bidirectional calls between satellite i and satellite j, $1 \leq i, j \leq 3$, regardless of whether the calls originated at satellite i or j. As an example, if $n_{12} = 1$, then there is one call using a one-way ISL channel from satellite 1 to satellite 2 and a one-way

ISL channel from satellite 2 to satellite 1. If $n_{11} = 1$, then there is a call between a customer under satellite 1 and a customer also under satellite 1, and two bidirectional UDL channels are used. Let λ_{ij} (respectively, $1/\mu_{ij}$) denote the arrival rate (resp., mean holding time) of calls between satellites i and j. Then, the three-satellite system in Figure 1 can be described by the Markov process:

$$\underline{n} \quad = \quad (n_{11}, n_{12}, n_{13}, n_{22}, n_{23}, n_{33}) \tag{1}$$

Let $\underline{1}_{ij}$ denote a vector with zeros for all random variables except random variable n_{ij} which is 1. The state transition rates for the Markov process are given by:

$$r(\underline{n}, \underline{n} + \underline{1}_{ij}) \quad = \quad \lambda_{ij} \quad \forall\, i, j \tag{2}$$

$$r(\underline{n}, \underline{n} - \underline{1}_{ij}) \quad = \quad n_{ij}\ \mu_{ij} \quad \forall\, i, j,\ n_{ij} > 0 \tag{3}$$

The transition (2) is due to the arrival of a call between satellites i and j, while the transition (3) is due to the termination of a call between satellites i and j.

Let Ω denote the state space for this Markov process. Due to the fact that some of the calls share common up-and-down and inter-satellite links, the following constraints are imposed on Ω:

$$2n_{11} + n_{12} + n_{13} \quad \leq \quad C_{UDL} \tag{4}$$

$$n_{12} + 2n_{22} + n_{23} \quad \leq \quad C_{UDL} \tag{5}$$

$$n_{13} + n_{23} + 2n_{33} \quad \leq \quad C_{UDL} \tag{6}$$

$$n_{12} \quad \leq \quad C_{ISL} \tag{7}$$

$$n_{13} \quad \leq \quad C_{ISL} \tag{8}$$

$$n_{23} \quad \leq \quad C_{ISL} \tag{9}$$

Constraint (4) ensures that the number of calls originating (equivalently, terminating) at satellite 1 is at most equal to the capacity of the up-and-down link of that satellite. Note that a call that originates and terminates within the footprint of satellite 1 captures two channels, thus the term $2n_{11}$ in constraint (4). Constraints (5) and (6) are similar to (4), but correspond to satellites 2 and 3, respectively. Finally, constraints (7)-(9) ensure that the number of calls using the link between two satellites is at most equal to the capacity of that link.

It is straightforward to verify that the Markov process for the three-satellite system shown in Figure 1 has a closed-form solution which is given by:

$$P(\underline{n}) = P(n_{11}, n_{12}, n_{13}, n_{22}, n_{23}, n_{33}) = \frac{1}{G} \frac{\rho_{11}^{n_{11}}}{n_{11}!} \frac{\rho_{12}^{n_{12}}}{n_{12}!} \frac{\rho_{13}^{n_{13}}}{n_{13}!} \frac{\rho_{22}^{n_{22}}}{n_{22}!} \frac{\rho_{23}^{n_{23}}}{n_{23}!} \frac{\rho_{33}^{n_{33}}}{n_{33}!}, \underline{n} \in \Omega \tag{10}$$

where G is the normalizing constant and $\rho_{ij} = \lambda_{ij}/\mu_{ij},\ i, j = 1, 2, 3$, is the offered load of calls from satellite i to satellite j. As we can see, the solution is the product of six terms of the form $\rho_{ij}^{n_{ij}}/n_{ij}!,\ i, j = 1, 2, 3$, each corresponding to one of the six different source/destination pair of calls. Therefore, it is easily generalizable to a k-satellite system, $k > 3$.

An alternative way is to regard this Markov process as describing a network of six M/M/K/K queues, one for each source/destination pair of calls between the three satellites. Since the satellites do not move, there are no hand-offs, and as a consequence customers do not move from one queue to another. Now, the probability that there are m customers in an M/M/K/K queue is given by the familiar expression $(\rho^m/m!)/\left(\sum_{l=0}^{K} \rho^l/l!\right)$, and therefore, the probability that there are $(n_{11}, n_{12}, n_{13}, n_{22}, n_{23}, n_{33})$ customers in the six queues is given by (10). Unlike previous studies, our model takes into account the fact that the six M/M/K/K queues are not independent, since the number of customers accepted in each M/M/K/K queue depends on the number of customers in other queues, as described by constraints (4)-(9).

Of course, the main concern in any product-form solution is the computation of the normalizing constant:

$$G = \sum_{\underline{n} \in \Omega} \frac{\rho_{11}^{n_{11}}}{n_{11}!} \frac{\rho_{12}^{n_{12}}}{n_{12}!} \frac{\rho_{13}^{n_{13}}}{n_{13}!} \frac{\rho_{22}^{n_{22}}}{n_{22}!} \frac{\rho_{23}^{n_{23}}}{n_{23}!} \frac{\rho_{33}^{n_{33}}}{n_{33}!} \tag{11}$$

where the sum is taken over all vectors $\underline{n}$ that satisfy constraints (4) through (9). A method to compute G is presented in [9]. Numerical experiments with this method indicate that it is limited to $k = 5$ satellites. That is, it takes an amount of time in the order of a few minutes to compute the normalizing constant G for 5 satellites. Thus, a different method is needed for analyzing realistic constellations of LEO satellites.

3 A Decomposition Algorithm for Satellite Constellations

We now present a decomposition method for calculating call blocking probabilities in a constellation of satellites. The constellation is decomposed into a series of sub-systems each consisting of at most three satellites. Each sub-system is analyzed separately using the exact solution described in the previous section. The results obtained from the sub-systems are then combined together using an iterative scheme in order to obtain a solution to the constellation as a whole.

As in the previous section, we will assume for the moment that the constellation of satellites is fixed over the earth, as in the case of geostationary satellites. That is, calls are not handed off from one satellite to another, and the call blocking probability due to hand-offs is zero. Therefore, the decomposition algorithm presented in this section can only calculate the call blocking probabilities of new calls. In the following section, we extend the algorithm to also calculate the call blocking probabilities due to hand-offs.

In order to explain how the decomposition algorithm works, let us consider a 16-satellite constellation with 4 orbits and 4 satellites per orbit, as shown in Figure 2. In the configuration of satellites that we study, we do not take into account the presence of the seam or the fact that satellites near the north and south pole have some of their links shut down. These two cases can be easily taken

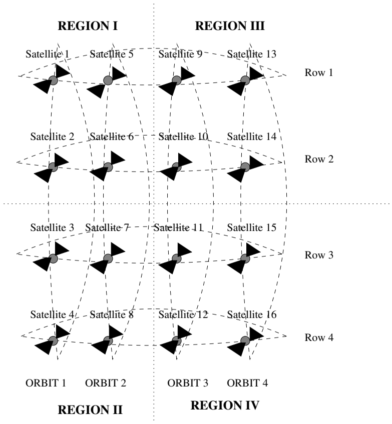


Fig. 2. 16-satellite constellation

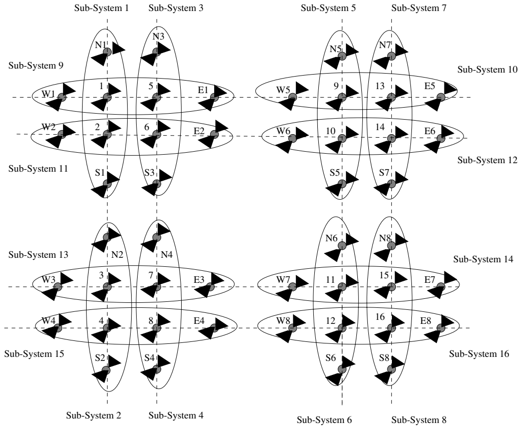


Fig. 3. Augmented sub-systems for constellation of Figure 2

onto account by simply changing the routing paths between pairs of satellites that are affected by the lack of links over the seam and near the poles.

The constellation is fixed over the earth, and we assume that each satellite in the first row has an intra-plane ISL to the satellite on the same orbit located in the bottom row. For instance satellite 1 communicates with satellite 4 via an intra-plane ISL. Likewise, satellites 5 and 8 are connected by an intra-plane ISL, and so on. Also, each satellite in the first column communicates via an inter-plane ISL with the satellite on the fourth column that is located on the same row. For instance, satellite 1 has an inter-plane link to satellite 13, and so on.

For the purposes of our decomposition algorithm, each orbit is divided into two sub-systems (shown in Figure 3). For instance, orbit 1 is divided into sub-system 1, consisting of satellites 1 and 2, and sub-system 2, consisting of satellites 3 and 4. Orbit 2 is divided into sub-system 3, consisting of satellites 5 and 6, and sub-system 4, consisting of satellites 7 and 8; likewise for orbits 3 and 4. Similarly, each row of four satellites in Figure 2 is divided into two sub-systems. The 16-satellite constellation is thus divided into 16 sub-systems as shown in Figure 3.

In order to analyze sub-system 1 in isolation, we need to have some information from other sub-systems. Specifically, we need to know the probability that a call originating at a satellite in sub-system 1 and terminating at a satellite in sub-system r, where $r > 1$, will be blocked due to lack of capacity in a link of any sub-system that it has to traverse, including sub-system r. Also, we need to know the number of calls that originate at other sub-systems and terminate in sub-system 1. Similar information is needed in order to analyze any other sub-system.

In view of this, each sub-system within an orbit is augmented to include two fictitious satellites, referred to as N and S. These two satellites are used to represent the aggregate traffic generated by other satellites and which flows into (or out of) the sub-system along links north or south of the sub-system, respectively. For instance, sub-system 1, shown in Figure 3, is augmented with fictitious satellites *N1* and *S1*. A call originating at satellite i, $i = 1, 2$ and terminating at satellite j, $j = 3, 4$ are represented in our sub-system by a call from satellite i to one of the fictitious satellites *N1* or *S1*. Depending upon i and j, this call may be routed differently. In our augmented sub-system, a call will be routed to *S1* if the shortest-path route passes through satellites south of the sub-system. A call will be routed to *N1* if the shortest-path route goes towards the north. In other words, satellite *N1* (respectively, *S1*) in the augmented sub-system is the destination satellite for all calls that originate in satellite i of sub-system 1 and are routed to satellite j located outside that sub-system in the clockwise (respectively counter-clockwise) direction.

This augmented sub-system captures the traffic outside the sub-system that travels on the same orbit, i.e., on intra-plane ISLs. In addition, we also have to consider traffic that uses inter-plane ISLs. For instance, let us consider again sub-system 1. A call originating at satellite 1 and terminating at satellite 6 will use the intra-plane ISL to satellite 2 and then the inter-plane ISL between satellites 2 and 6. In order to account for traffic traversing inter-plane ISLs, we also decompose each row of satellites into two sub-systems, each consisting of two satellites. For instance, the first row of satellites is divided into sub-system 9, consisting of satellites 1 and 5, and sub-system 10, consisting of satellites 9 and 13. The 16-satellite constellation is thus divided into an additional 8 sub-systems, as shown in Figure 3. Each sub-system is augmented to include two fictitious satellites, referred to as E and W. As before, the fictitious E and W satellites are used to represent the aggregate traffic generated by other satellites and which flows into (or out of) the sub-system along links east or west of the sub-system, respectively. For instance, a call originating at say satellite i, $i = 1, 5$, and terminating at satellite j, $j = 9, 13$, will be represented in our sub-system 9 as a call from i to either $E1$ or $W1$, depending upon the shortest-path route of the call. As another example, consider a call between satellites 5 and 11. Using shortest-path routing, this call is routed through satellites 9 and 10. Within the augmented sub-system 9 this particular is represented as a call between satellite 5 and fictitious satellite $E1$.

In order to analyze the augmented sub-systems in Figure 3, we introduce the *effective* arrival rates $\hat{\lambda}_{ij}$, including rates $\hat{\lambda}_{i,N}$,$\hat{\lambda}_{i,S}$ (or $\hat{\lambda}_{i,E}$, $\hat{\lambda}_{i,W}$), within each sub-system. The effective rate $\hat{\lambda}_{ij}$ captures the rate of calls between satellite i and satellite j, *as seen from within this sub-system*. In particular, the effective rate $\hat{\lambda}_{i,N}$ (or any other rate involving any of the other fictitious satellites S, E or W) captures the rate of calls originating at satellite i and leaving the sub-system over an ISL that goes through the fictitious satellite N.

Based on this decomposition, computing the blocking probability of a call depends on whether or not the originating and terminating satellites of the call are within the same sub-system. In the former case, the blocking probability is computed directly as a byproduct of solving the sub-system in isolation. In the latter case, the blocking probability is computed by taking into account all the sub-systems in the call's path. Returning to Figure 3, a call originating at satellite 1 and terminating at satellite 6 will be analyzed in two steps. At the first step, it is a call within sub-system 1 between satellites 1 and 2. This call then leaves this sub-system from satellite 2 and it is analyzed using sub-system 11. From the point of view of sub-system 11, this is a call between satellites 2 and 6. As another example, analyzing a call between satellite 1 and satellite 8 involves three sub-systems. Within sub-system 1, it is viewed as a call between satellite 1 and (fictitious) satellite $N1$. In sub-system 2, it is considered a call between (fictitious) satellite $S2$ and satellite 4. Finally, in sub-system 15, it is a call between satellites 4 and 8.

We now illustrate the decomposition algorithm using the 16 satellite constellation shown in Figure 3. Initially, we solve sub-system 1 in isolation. This system in isolation is described by the following Markov process:

$$\underline{n} = (n_{11}, n_{12}, n_{1N_1}, n_{1S_1}, n_{22}, n_{2N_1}, n_{2S_1}) \tag{12}$$

We solve sub-system 1 exactly using the approach described in the previous section. The arrival rates used in the solution are the effective arrival rates $\hat{\lambda}_{1,N1}, \hat{\lambda}_{1,S1}, \hat{\lambda}_{1,2}, \hat{\lambda}_{2,N1}$, and $\hat{\lambda}_{2,S1}$. Efective rate $\hat{\lambda}_{1,N1}$ is obtained using expression (13); the other effective rates (for this or other sub-systems) are obtained from similar expressions which can be found in [10].

$$\begin{aligned}\hat{\lambda}_{1,N1} = {} & (1 - p_{4,S_2})\lambda_{1,4} + (1 - p_{4,S_2})(1 - p_{4,8})\lambda_{1,8} + (1 - p_{4,S_2})(1 - p_{4,8}) \\ & \times(1 - p_{W_8,12})\lambda_{1,12} + (1 - p_{4,S_2})(1 - p_{4,E_4})(1 - p_{W_8,16})\lambda_{1,16}\end{aligned} \tag{13}$$

We now explain expression (13) in more detail. Note that, in this expression, quantities p_{ij} represent the probability that a call between two satellites traveling through the path segment (i, j) in another sub-system will be blocked due to the lack of capacity in that segment.

Consider expression (13) for effective rate $\hat{\lambda}_{1,N1}$ which represents the rate of calls originating at satellite 1 and leaving the sub-system over ISL 1-4 in Figure 2. Because of the shortest path routing we consider here, these are calls terminating at satellites 4, 8, 12, and 16. Consequently, the right-hand side of (13) consists of four terms, one for calls terminating at each of these four satellites. The first term in (13), $(1-p_{4,S_2})\lambda_{1,4}$, represents the effective arrival rate of calls between satellites 1 and 4, as seen by sub-system 1. This effective rate represents the fraction of calls between satellites 1 and 4 not blocked in sub-system 2 between satellites 4 and S_2, and is given by the product of the arrival rate $\lambda_{1,4}$ of new calls between satellites 1 and 4 times the probability that a call is not blocked between satellite 4 and (fictitious) satellite $S2$) in sub-system 2. The second term is obtained similarly by accounting for all the sub-systems in the shortest path between satellites 1 and 8. A call between satellites 1 and 8 may be blocked either in sub-system 2, between satellites 4 and S_2, or in sub-system 15, between satellites 4 and 8. Therefore, the effective arrival rate for a call between satellites 1 and 8 as seen by sub-system 1 is $(1-p_{4,S_2})(1-p_{4,8})\lambda_{1,8}$. This expression gives us the proportion of calls that are not blocked in sub-systems 2 and 15. The third term $(1-p_{4,S_2})(1-p_{4,8})(1-p_{W_8,12})\lambda_{1,12}$ provides the effective arrival rate between satellites 1 and 12. This expression gives us the proportion of the traffic that is not blocked between satellites 4 and S_2, 4 and 8, and W_8 and 12. The last term of $\lambda_{1,N1}$ is similar with the previous term except it accounts for the sub-systems along the shortest path to satellite 16.

Equations similar to (13) are used to solve sub-system 1, as well as other sub-systems, in isolation. The values of quantities p_{ij} are updated at each iteration, and represent our best estimate for the value of the corresponding blocking probability at the beginning of the iteration. For the first iteration, we use $p_{ij}^{(0)} = 0$, for all i and j. During the h-th iteration, each sub-system is solved in isolation using the blocking probabilities $p_{ij}^{(h-1)}$ computed during the previous iteration. As a result of the solution to the sub-system a new set of values $p_{ij}^{(h)}$ for the blocking probabilities are obtained, and these are used in the next iteration. This iterative procedure continues until the blocking probabilities converge.

Any constellation with a large number of satellites can be decomposed in a similar manner into a number of sub-systems, each of 3 or fewer satellites.

4 Modeling Hand-Offs

4.1 Earth-Fixed Coverage

In a LEO satellite constellation with earth-fixed coverage, time is divided in intervals of length T such that, during a given interval, each satellite serves a certain cell by continuously redirecting its beams. At the end of each interval, i.e., every T time units, all satellites simultaneously redirect their beams to serve the next footprint along their orbit, and they also hand-off currently served calls to the next satellite in the orbit. Therefore, hand-off events are periodic with a period of T time units, and hand-offs take place in bulk at the end of each period. There is no call blocking due to hand-offs, since, at each hand-off event a satellite passes its calls to the one following it and simply inherits the calls of the satellite ahead of it. Within each period T, the system can be modeled as one with no hand-offs. Given that the period T is equal to the orbit period (approximately 100 minutes) divided by the number of satellites at each orbit, we can assume that the system reaches steady state within the period, and thus, the initial conditions (i.e., the number of calls inherited by each satellite at the beginning of the period) do not affect its behavior. Furthermore, from the point of view of an observer on the earth, this system appears to be as if the satellites are permanently fixed over their footprints. Hence, we can use the decomposition algorithm presented above to analyze this system.

4.2 Satellite-Fixed Coverage

Consider now satellite-fixed cell coverage. As a satellite moves, its footprint on the earth (the cell served by the satellite) also moves with it. As customers move

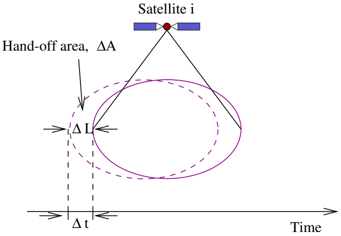


Fig. 4. Calculation of the hand-off probability

out of the footprint area of a satellite, their calls are handed off to the satellite following it from behind. In order to model hand-offs in this case, we make the assumption that potential customers are uniformly distributed over the earth. Clearly, this assumption is an approximation.

Let A denote the area of a satellite's footprint and v denote a satellite's speed. As a satellite moves around the earth, within a time interval of length Δt, its footprint will move a distance of ΔL, as shown in Figure 4. Calls involving customers located in the part of the original footprint of area ΔA (the hand-off area) that is no longer served by the satellite are handed off to the satellite following it. Let $\Delta A = A\beta\Delta L$, where β depends on the shape of the footprint. Because of the assumption that active customers are uniformly distributed over the satellite's footprint, the probability q that a customer will be handed off to the next satellite along the sky within a time interval of length Δt is

$$q = \frac{\Delta A}{A} = \beta\Delta L = \beta v\Delta t \tag{14}$$

Define $\alpha = \beta v$. Then, when there are n customers served by a satellite, the *rate* of hand-offs to the satellite following it will be αn.

Single Sub-System. Let us first return to the 3-satellite orbit (see Figure 1) and introduce hand-offs. This system can be described by a continuous-time Markov process with the same number of random variables as the no-hand-offs model of Section 2 (i.e., $n_{11}, \cdots, n_{33}$), the same transition rates (2) and (3), but with a number of additional transition rates to account for hand-offs. We will now derive the transition rates due to hand-offs.

Consider calls between a customer served by satellite 1 and a customer served by satellite 2. There are n_{12} such calls serving $2n_{12}$ customers: n_{12} customers on the footprint of satellite 1 and n_{12} on the footprint of satellite 2. Consider a call between customer A and customer B, served by satellites 1 and 2, respectively. The probability that customer A will be in the hand-off area of satellite 1 but B will not be in the hand-off area of satellite 2 is $q(1-q) = q - q^2$. From (14), we have that $\lim_{\Delta t \to 0} \frac{q^2}{\Delta t} = 0$, so the rate at which these calls experience a hand-off from satellite 1 to satellite 3 that follows it is αn_{12}. We have:

$$r(\underline{n}, \underline{n} - \underline{1}_{12} + \underline{1}_{23}) = \alpha n_{12}, \quad n_{12} > 0 \tag{15}$$

Similarly, the probability that customer B will be in the hand-off area of satellite 2 but A will not be in the hand-off area of satellite 1 is $q(1-q) = q-q^2$. Thus, the rate at which these calls experience a hand-off from satellite 2 to satellite 1 that follows it is again αn_{12}:

$$r(\underline{n}, \underline{n} - \underline{1}_{12} + \underline{1}_{11}) \quad = \quad \alpha n_{12}, \quad n_{12} > 0 \tag{16}$$

On the other hand, the probability that both customers A and B are in the hand-off area of their respective satellites is q^2, which, from (14) is $o(\Delta t)$, and thus simultaneous hand-offs are not allowed.

The transition rates involving the other four random variables in the state description (1) can be derived using similar arguments, and can be found in [10].

From the queueing point of view, this system is a queueing network of M/M/K/K queues as described in Section 2, where customers are allowed to move between queues. This queueing network has a product-form solution similar to (10). Let γ_{ij} denote the total arrival rate of calls between satellites i and j, including new calls (at a rate of λ_{ij}) and hand-off calls (at an appropriate rate). The values of γ_{ij} can be obtained by solving the traffic equations for the queueing network. Let $\nu_{ij} n_{ij}$ be the departure rate when there are n_{ij} of these calls, including call termination (at a rate of $\mu_{ij} n_{ij}$) and call hand-off (at a rate of $2\alpha n_{ij}$). Define $s_{ij} = \gamma_{ij}/\nu_{ij}$. The solution for this queueing network is:

$$P(\underline{n}) = P(n_{11}, n_{12}, n_{13}, n_{22}, n_{23}, n_{33}) = \frac{1}{G} \frac{s_{11}^{n_{11}}}{n_{11}!} \frac{s_{12}^{n_{12}}}{n_{12}!} \frac{s_{13}^{n_{13}}}{n_{13}!} \frac{s_{22}^{n_{22}}}{n_{22}!} \frac{s_{23}^{n_{23}}}{n_{23}!} \frac{s_{33}^{n_{33}}}{n_{33}!} \tag{17}$$

which is identical to (10) with s_{ij} in place of ρ_{ij}. Therefore, the exact solution od Section 2 is applicable to this new queueing network as well.

Constellation of Satellites. To analyze a constellation of satellites with hand-offs, we use the algorithm presented in Section 3. The main difference is that, instead of using the arrival and departure rates for new calls, λ_{ij} and μ_{ij}, respectively, we use the rates γ_{ij} and ν_{ij} which account for both new and hand-off calls. The latter are obtained by solving the traffic equations for the queueing network. Therefore, our analysis of a constellation follows the steps below:

1. The constellation is modeled as a queueing network of M/M/K/K queues, where each queue represents the number of calls between a pair of satellites (i, j) (no hand-offs case). A number of constraints, similar to (4)-(9), are imposed to account for the fact that some calls share common links.
2. In order to model hand-offs, we introduce additional transitions of customers moving from one queue to another.
3. We solve exactly the traffic equations of the queueing network resulting from Step 2 to obtain the new arrival rates.
4. We apply the decomposition algorithm described in Section 3 using the arrival rates from Step 3.

Solving the traffic equations is computationally expensive, taking time $O(N^3)$, where N is the number of states in the Markov process. The number N of

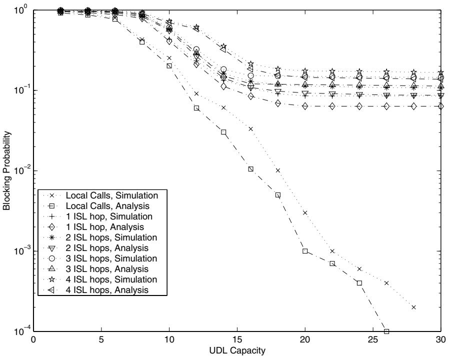


Fig. 5. Call blocking probabilities for 16 satellites with hand-off, uniform pattern

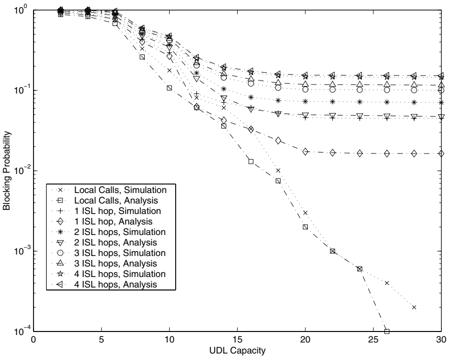


Fig. 6. Call blocking probabilities for 16 satellites with hand-off, locality pattern

states, in turn, is exponential in the number K of satellites. To decrease the complexity, we have developed an approximate way to solve the traffic equations in a distributed manner; a detailed description can be found in [10].

5 Numerical Results

In this section we verify the accuracy of the decomposition algorithm by comparing to simulation results. 95% confidence intervals were estimated by the method of replications. The number of replications is 30, with each simulation run lasting until each source/destination pair of call has at least 15,000 arrivals. For the approximate results, the decomposition algorithm terminates when all call blocking probability values have converged within 10^{-6}.

We obtained results using three different traffic patterns: a uniform traffic pattern, one based on the notion of traffic locality, and a hot spot pattern (for details, refer to [10]). We consider a constellation of 16 satellites with four orbits and four satellites per orbit as shown in Figure 2. Each satellite has four ISLs; two within the same orbit and two with neighboring orbits.

Figures 5-7 plot the call blocking probability (for new and hand-off calls) against the capacity C_{UDL} of up-and-down links, when the arrival rate $\lambda = 5$ and the capacity of inter-satellite links $C_{ISL} = 10$, for the three traffic pattern. We note that there is a good agreement between the analytical results and the simulation. Overall, the analytical curves track the simulation curves accurately, indicating that the decomposition algorithm can be used to predict the call blocking performance of a LEO satellite constellation accurately and efficiently.

6 Concluding Remarks

We have presented an analytical model for computing call blocking probabilities in LEO satellite networks. We have developed an algorithm for decomposing the

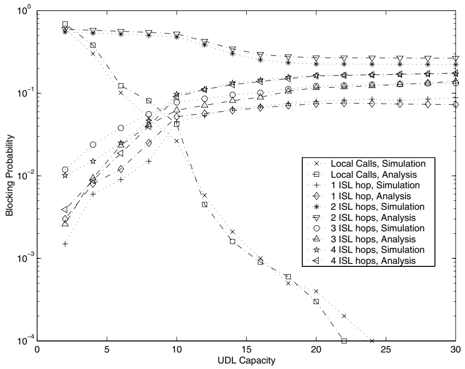


Fig. 7. Call blocking probabilities for 16 satellites with hand-off, hot-spot pattern

constellation into smaller sub-systems, each of which is solved in isolation exactly. The individual solutions are combined using an iterative scheme. We have also shown how our approach can capture blocking due to hand-offs for both satellite-fixed and earth-fixed coverage. We have demonstrated through numerical examples that the analytical results are in good agreement with simulation.

References

1. F. Dosiere, *et al.* A model for the handover traffic in low earth-orbiting satellite networks for personal communications. In *IEEE Globecom*, 1993.
2. A. Ganz, Y. Gong, and B. Li. Performance study of low earth orbit satellite systems. *IEEE Trans. Commun.*, 42(2/3/4), February/March/April 1994.
3. V. Obradovic and S. Cigoj. Performance evaluation of prioritized handover management for LEO mobile satellite systems with dynamic channel assignment. In *Global Telecom. Conf.*, vol. 1a, 1999.
4. G. Pennoni and A. Ferroni. Mobility management in LEO/ICO satellite systems: Preliminary simulation results. In *PIMRC*, pages 1323–1329, 1994.
5. E. D. Re, *et al.* Different queueing policies for handover requests in low earth orbit mobile satellite systems. *IEEE Trans. Veh. Tech.*, 48(2):448–458, March 1999.
6. J. Restrepo and G. Maral. Guaranteed handover (GH) service in a non-geo constellation with “satellite-fixed-cell” (SFC) systems. In *Int. Mobile Sat. Conf.*, 1997.
7. G. Ruiz, T. L. Doumi, and J. G. Gardiner. Teletraffic analysis and simulation of mobile satellite systems. *IEEE Trans. Veh. Tech.*, 47(1):311–320, February 1998.
8. P. J. Wan, V. Nguyen, and H. Bai. Advanced handovers arrangement and channel allocation in LEO satellite networks. In *Global Telecom. Conf.*, vol. 1a, 1999.
9. A. Zaim, G. Rouskas, and H. Perros. Computing call blocking probabilities in LEO satellite networks: The single orbit case. *IEEE Trans. Veh. Techn.*, 51(1), Jan 2002.
10. A. Halim Zaim. *Computing Call Blocking Probabilities in LEO Satellite Networks.* PhD thesis, North Carolina State University, Raleigh, NC, August 2001.

Gateway Architecture for DVB-RCS Satellite Networks

Antonio Pietrabissa[1] and Cristiana Santececca[2]

University of Rome “La Sapienza”
Dipartimento di Informatica e Sistemistica (DIS)
[1]pietrabissa@dis.uniroma1.it, [2]cristiana.santececca@tin.it

Abstract. The introduced gateway architecture for DVB-based geostationary satellite networks has been developed for the European Community project “GOECAST” (multiCAST over GEOstationary satellites) and aims at supporting the real-time (RT) traffic feeding the Gateway Earth Stations (GES). GESs provide the access to the satellite network to a large number of RT flows, whose delay constraints have to be guaranteed without affecting the lower priority traffic. A tight control on the queuing delays is obtained by the proposed scheduling scheme, while the proposed buffering scheme allows the scalability requirement to be met by means of a proper traffic aggregation criterion. Finally, the ‘stolen slot’ concept is introduced: whenever the RT traffic transmission rate is lower than the nominal rate, the ‘stolen-slot’ procedure allows the utilisation of the leftover bandwidth, considering also the multicast issues. Simulations have been performed with the OPNET tool to test the effectiveness of the proposed schemes and algorithms.

1 The GEOCAST Project

The GEOCAST scenario consists of a geostationary (GEO) satellite network with an on-board packet-switch, a Network Control Centre (NCC), in charge of several key tasks relevant to resource management and connection handling, several User Earth Stations (UES), which provide the access to few User Terminals (UT), and a limited number of Gateway Earth Stations (GES), which provide the access to a large population of users and to backbone networks. The GES uplink access is TDM (Time Division Mutiplexing): the uplink capacity is divided into time-slots; each time-slot is capable of transporting one packet. The UES uplink, conversely, is MF-TDMA (Multi Frequency Time-Division Multiple Access), so that the UESs are capable of sharing the uplink capacity. Fig. 1 shows the GEOCAST scenario.

While each GES has the exclusive use of the TDM uplink frames, the downlink capacity is shared among several GESs and UESs. Thus, the NCC, which manages the network resources, grants the requested rates of the GES connections; in particular, the Connection Admission Control (CAC) grants the requested capacity after the connection set-up for the connection life-time, while the Bandwidth-on-Demand (BoD) scheme (presented in [1]) grants some capacity in response to the capacity requests.

The protocol stack of the GEOCAST project is compliant with the DVB-RCS (Return Channel via Satellite-Digital Video Broadcasting) standard ([2], [3]). In particu-

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 802-813, 2002.

lar, it uses the DVB-RCS ATM (Asynchronous Transfer Mode) format on the uplink and the DVB-S MPEG2 format on the downlink ([4]). Without explaining the overall protocol stack in detail, this means that the IP traffic entering the Gateway is mapped onto ATM connections, which are transported through the DVB priority traffic classes ([3]): real-time (RT), jitter tolerant (JT) and best effort (BE).

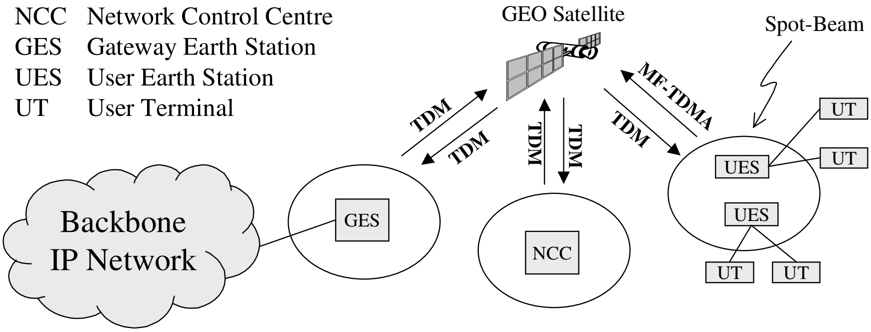


Fig. 1. GEOCAST scenario

This paper deals with the RT priority traffic, which is used to transport delay sensitive ATM connections; the ATM connections are characterised by the declared Quality of Service (QoS) parameters, as the Cell Transfer Delay (CTD) and the Cell Delay Variation (CDV), and by the declared traffic parameters, as the Peak Cell Rate (PCR). The CTD is the maximum acceptable delay perceived by the ATM cells from the source GES to the destination GES or UES; the CDV is the maximum allowed difference among the delays perceived by the ATM cells of the connection (jitter); the PCR is the maximum transmission rate. The ATM connections feeding the GES are subject to traffic shaping and policing through the Usage Parameter Control (UPC) blocks [5].

Furthermore, the GEOCAST network supports multicast traffic, which is mapped onto ATM one-to-many connections. For instance, a multicast session involving n users is mapped onto n one-to-many connections, each one with one sender and (n-1) receivers. The GEOCAST on-board switch is capable of duplicating the received packets and of forwarding them towards the proper downlinks.

In Section 2, the problems related to the satellite networks are highlighted; in Section 3, a Gateway architecture is proposed; in Section 4, the ‘stolen slot’ concept is defined taking into account the multicast issues; Section 5 presents the simulations results; finally, in Section 6 the conclusions are drawn.

2 Real-Time (RT) Traffic over Satellite Networks

The uplink TDM frame of the GESs is divided into frames. Each frame consists of a number N of time-slots and its duration is equal to T_{FRAME}. Each time-slot is capable of transporting one ATM cell (hereinafter referred to as packet). In the GEOCAST project, $N = 1136$, $T_{FRAME} = 53$ ms and each time-slot has a capacity $C_{SLOT} = 8$ kbps (i.e., the capacity equivalent to 1 time-slot per frame is 8 kbps).

The RT traffic has tight delay constraints, thus it cannot avail of the BoD mechanism, since the time required by the NCC to fulfil the bandwidth requests is equal to the Round Trip Delay (RTD) of the geostationary networks, which is about 500 ms and is likely to be greater than the CTD of the RT connections. As a consequence, the network cannot react to the rate variations and, therefore, during the connection set-up, the CAC in the NCC must reserve a capacity equal to the Peak Cell Rate (PCR) for the connection lifetime. In the one-to-many connection case, the PCR capacity must be reserved onto each involved downlink. Finally, the GES must check whether the required PCR is less than its available capacity, i.e., the uplink capacity minus the capacity already reserved for other active connections.

Usually ([6]-[9]), the RT connections are associated to one or more logical channels, that is one or more time-slot per frame are assigned to the connection.

This solution has a problem of over-allocation, since where the number of time-slots N_{PCR} required to map the connection is given by the following equation: $N_{PCR} = \lceil PCR / C_{SLOT} \rceil$. On average, each connection needs 0.5 C_{SLOT} = 4 kbps more than its PCR. In addition, this scheme requires that each ATM connection – named Virtual Channel Connection (VCC) – has its own buffer in the GES. This buffering scheme causes a scalability problem for the GESs, because of the large number of connections the GESs have to support. Furthermore, the queues in the buffers, and thus the CDV of the connections, depend on the allocation policy and on the declared PCR.

As a mater of fact, if the channels are contiguous, the minimum queuing delay is zero, if the packet arrives in the buffer when the time-slot is available, while the maximum queuing delay is about T_{FRAME}, if the packet arrives in the buffer just after the reserved time-slot. Thus, in this case, the CDV is about T_{FRAME}. Fig. 2 a) shows an example of this mapping policy. Note that the arrival time $t_a(j)$ of the j^{th} packet is equal to the beginning time of the time-slot used to transmit the packet itself; thus, the queuing delay perceived by the j^{th} packet is zero. On the other hand, the arrival time $t_a(j+1)$ of the $(j+1)^{th}$ packet is slightly greater than the beginning time of the assigned time-slot; thus, the packet can be transmitted only in the beginning of the following frame. Thus, the queuing delay perceived by the $(j+1)^{th}$ packet is about T_{FRAME}.

If the GES allocates the time-slots regularly over the frame, it manages to limit the CDV to T_{FRAME} / N_{PCR}. Fig. 2 b) shows an example of this mapping policy, in which the PCR of the connection is such that N_{PCR} = 3 time-slots per frame. Note that the arrival time $t_a(j)$ of the j^{th} packet is slightly greater than the beginning time of the assigned time-slot; thus, the packet can be transmitted only via the successive time-slot. Thus, the queuing delay perceived by the j^{th} packet is about T_{FRAME} / N_{SLOT}. On the other hand, the arrival time $t_a(j+1)$ of the $(j+1)^{th}$ packet is equal to the beginning time of the time-slot used to transmit the packet itself; thus, the queuing delay perceived by the $(j+1)^{th}$ packet is zero. Note, however, that a regular allocation might be impossible because of already allocated time-slots; in this case, the connection suffers from a CDV > T_{FRAME} / N_{PCR}.

In conclusion, the drawbacks of this architecture are: i) over-allocation of uplink resources; ii) scalability due to the per-VCC buffering scheme; iii) obtainable CDV limited by T_{FRAME} or PCR iv) possible connection denial due to the impossibility of mapping the connection properly. In the following Section, the proposed architecture is presented, which overcomes the above mentioned problems.

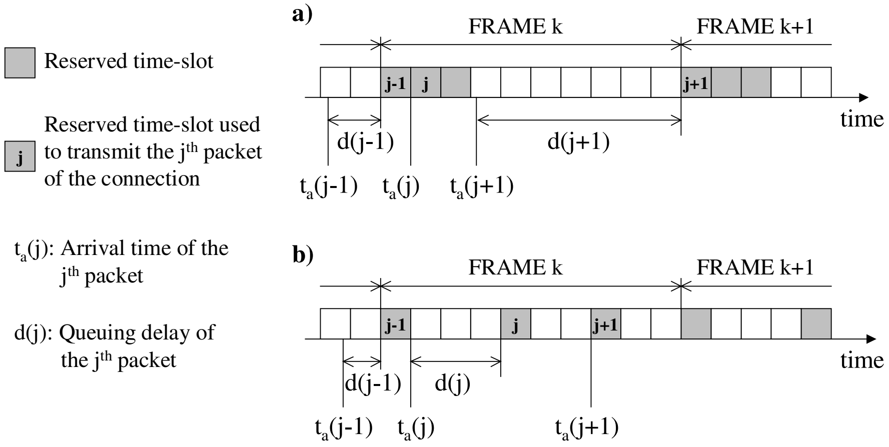

Reserved time-slot
j
Reserved time-slot used to transmit the jth packet of the connection
ta(j): Arrival time of the jth packet
d(j): Queuing delay of the jth packet
a)
FRAME k
FRAME k+1
j-1
j
j+1
time
d(j-1)
d(j+1)
ta(j-1)
ta(j)
ta(j+1)
b)
FRAME k
FRAME k+1
j-1
j
j+1
time
d(j-1)
d(j)
ta(j-1)
ta(j)
ta(j+1)

Fig. 2. Queuing delays associated to the frame mapping policies

3 Proposed Gateway Architecture

The proposed GES architecture aims at allowing the transmission of the RT connections to be independent of the uplink frame structure and at increasing the scalability.

This latter issue is addressed through an appropriate traffic aggregation policy. First of all, the RT VCC of the GES directed towards the same downlink (or downlinks in the case of the one-to-many connections) are aggregated into Virtual Path Connections (VPC); consequently, the VPC are defined by the couple {Source GES, Destination Downlink(s)}. Then, within each VPC, the VCCs are aggregated on the basis of their QoS, so that the VCCs characterized by similar CDV requirements are collected in the same buffer. This aggregation policy allows a great reduction of the buffer number, and, as a consequence, of the complexity of the scheduling algorithm (as it will be explained later), without affecting the QoS perceived by the distinct VCCs. As a matter of fact, in order to assure a certain CDV the GES must limit the queuing delay; if the queuing delay of the QoS buffer is kept under the most stringent CDV requirements among the ones of its VCC, the CDV target is met by each VCC.

In order to render the transmission of the RT packets independent of the frame structure of the uplink, the following considerations are taken into account:

- The transmission rates of the RT connections feeding the GES are controlled by the UPC blocks, so that the packet rate of each connection cannot exceed the PCR.
- The RT connections are subject to the CAC, which assure that an uplink capacity equal to the PCR is always available.

Thus, the RT packets feeding the QoS buffers have always the matching available capacity on the uplink, and, because of the greater priority of the RT traffic with respect to the non real-time (NRT) traffic, should be transmitted as soon as possible.

Therefore, the proposed architecture features a two-levels scheduling scheme (see Fig. 3):

1. The first scheduling level consists of a strict priority scheduler, which decides whether a RT packet or a NRT packet must be transmitted; this scheduler transmits NRT packets only if the RT buffer queues are empty.
2. The second scheduling level consists of a RT and a NRT schedulers, which decides which packets to send among the ones waiting in the RT and NRT queues, respectively.

This paper is focused onto the RT traffic, thus only the RT scheduler will be examined. The Earliest Deadline First (EDF) scheduler seems to be adequate, because of the following reasons:

- The queuing delay is the only criterion, on the basis of which the RT packets must be scheduled;
- The maximum queuing delay allowed to a certain buffer *k*, $d_{MAX}(k)$, is known, since it given by the minimum CDV requirement of the VCCs aggregated in the buffer *k*;
- The EDF is known to to be optimum in the sense that it minimizes the total probability of exceeding the deadlines for all streams ([10], [11]).

The EDF scheduler computes the deadlines of the packets by adding $d_{MAX}(k)$ to the arrival time of the packets feeding the buffer *k*, and then it schedules the packet with the earliest deadline.

Thanks to the double aggregation of the VCCs in QoS classes and in VPCs, a two-levels scheduler can be implemented: the first EDF scheduler decides which packet to sent among the ones in the head of the QoS queues of the single VPCs, and has complexity $O(\log N_{QoS})$, where N_{QoS} is the number of QoS classes defined within the RT class; the second scheduler decides decides which packet to sent among the ones selected by the first level schedulers, and has complexity $O(\log N_{VPC})$, where N_{VPC} is the number of VPCs.

Fig. 3 shows the proposed GES architecture.

While the proposed architecture succeeds in enhancing the scalability and in overcoming the problems related to the connection mapping, it has a drawback. During the j^{th} frame, the NRT scheduler is allowed to transmit packets towards the k^{th} downlink up to a certain capacity $C_{NRT}(j,k)$, which is granted by the NCC (by the CAC and by the BoD protocol). Since it is not likely that the RT sources transmit continuously at the PCR, the capacity available to the RT traffic on the k^{th} downlink which is left unused should increase $C_{NRT}(j,k)$. In the reference architecture, the RT connections are explicitly mapped onto the uplink frame; thus, if the RT connection buffer is empty in the beginning of an assigned time-slot, it can straightforwardly used by the NRT traffic. This simple procedure cannot be followed in the proposed architecture case. In the following Section, a procedure suitable for the proposed architecture is presented.

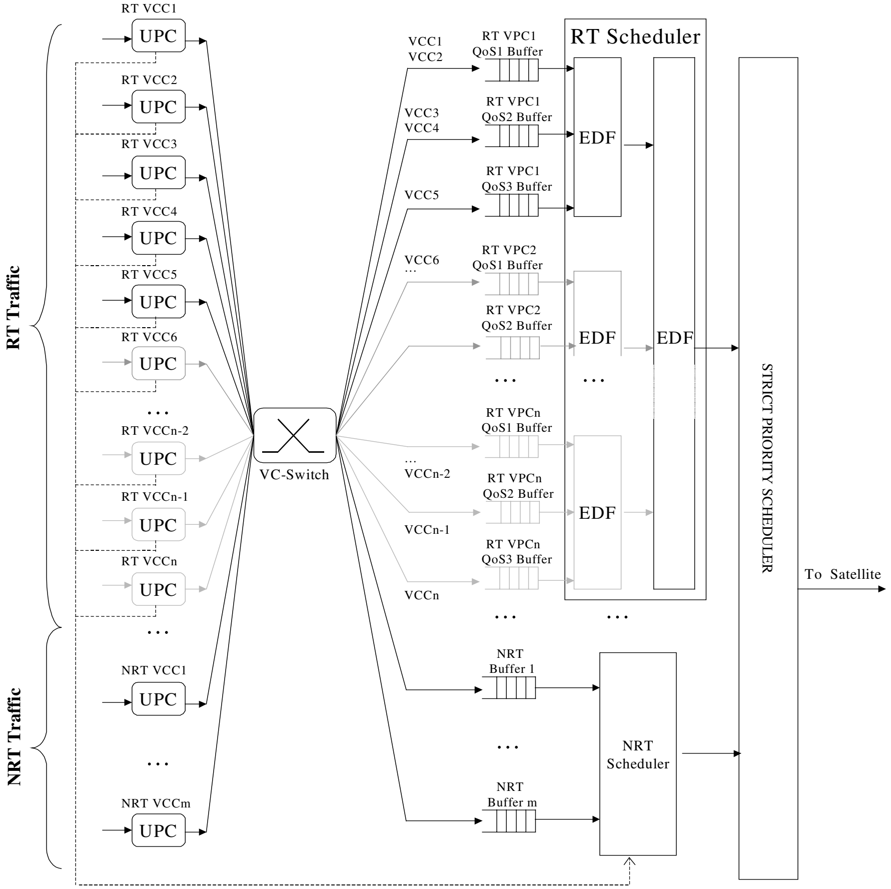


Fig. 3. Proposed gateway architecture

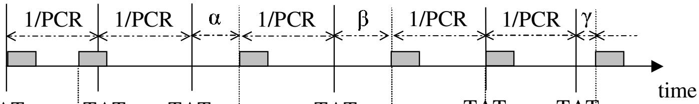
4 'Stolen-Slot' Procedure for Unicast and Multicast Traffic

Every time the RT connections transmit with a lower rate than the declared one, some capacity is left unused; when this capacity is equal to the time-slot capacity C_{SLOT}, one NRT packet can be served, and the time-slot is considered as 'stolen'. Two problems arise: i) how to compute the 'stolen' capacity, ii) how to assign the 'stolen' time-slots.

In order to identify whether a time-slot can be 'stolen', the proposed algorithm makes use of the UPC blocks of the GESs ([5]). Each VCC is characterised by the peak transmission rate PCR. When the first packet of the VCC arrives at $t = t_a(1)$, the associated UPC block computes the Theoretical Arrival Time (TAT) of the second packet: $TAT_2 = TAT_1 + 1/PCR$, where $TAT_1 = 0$. Then, when the second packet arrives at $t = t_a(2)$, the UPC block performs the following actions (neglecting the policing function): i) if $t_a(2) \geq TAT_2$, $TAT_3 = t_a(2) + 1/PCR$; ii) if $t_a(2) \leq TAT_2$, $TAT_3 = TAT_2 +$

1/*PCR*. Similarly, on the basis of TAT_k, when the k^{th} packet arrives at $t_a(k)$ the UPC computes $TAT_{(k+1)}$.

The ‘stolen-slots’ algorithm is based on the fact that when a cell arrives at time $t_a(k)$ after its *TAT(k)*, then the next *TAT(k+1)* is delayed. In particular, *TAT(k+1)* is delayed by $[t_a(k) - TAT(k)]$. Since the TATs cannot be anticipated, this time is ‘stolen’. When the sum of the stolen times is equal to 1/*PCR*, then 1 time-slot can be ‘stolen’, and the UPC block communicates it to the NRT scheduler (see Fig. 3). Fig. 4 shows an example run of the algorithm..



TAT	TAT_1	TAT_2	TAT_3	TAT_4	TAT_5	TAT_6
Arrival Time	$t_a(1)$	$t_a(2)$	$t_a(3)$	$t_a(4)$	$t_a(5)$	$t_a(6)$
Stolen Time	0	0	α = 0.5/PCR	β = 0.6/PCR	0	γ = 0.2/PCR
Tot. Stolen Time	0	0	0.5/PCR	1.1/PCR	0.1/PCR	0.3/PCR
Stolen Slots	0	0	0	1	1	1

Fig. 4. Example of the stolen slot algorithm

After that a time-slot is stolen, it has to be assigned to the proper connection. We recall that, while each GES has the exclusive use of the TDM uplink frames, the downlink capacity is shared among several GESs and UESs by the NCC. Thus, in the beginning of each frame, the GES is allowed to transmit a known capacity towards each downlink. In the case of one-to-many connections, the capacity is reserved by the NCC on each downlink involved by the connection, as shown in Fig. 5, in which the one-to-many VPC0 has one packets to send during the j^{th} frame and is allowed to transmit two packets (i.e., it has two allocated time-slots).

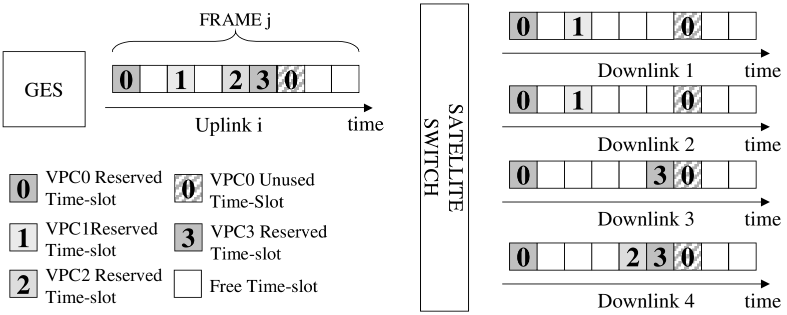


Fig. 5. Uplink and downlink reserved capacity

Let Δ be the set of the downlinks involved by the VPC the stolen slot belongs to. The problem is to decide which VPC should steal the time-slot:

1. First of all, the GES must check which are the eligible VPCs on the basis of the following rules: i) the VPC buffer in the GES must not be empty; ii) the set *D(j)* of the downlinks involved by the VPCj is a subset of Δ: $D(j) \subseteq \Delta$.
2. Then, the eligible VPCs are ordered with respect to i) the priority traffic and ii) the number of involved downlinks, i.e., the cardinality of *D(j).*
3. Finally, the time-slot is given to the first VPC of the list.

Note that, in the one-to-many case, the time-slot might be stolen by a certain VPCj, which does not involve all the downlinks, on which the capacity has been reserved. Let be *D(j)* be the set of downlinks involved by VPCj. By updating Δ in the following manner: Δ' = Δ \ *D(j)*, if another uplink time-slot is available to the GES, the above mentioned procedure can be repeated.

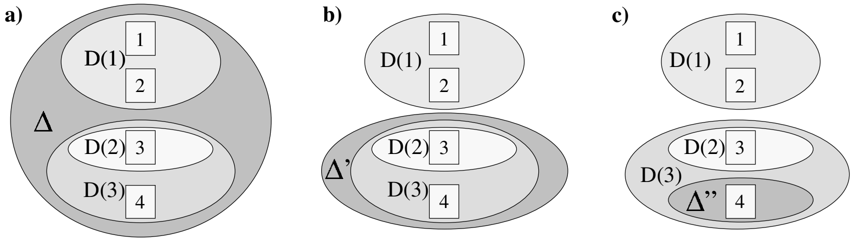
In conclusion, the 'stolen slot' procedure allows the GES to transmit *m* cells – belonging to one connection each – towards *n* downlinks ($n \geq m$), assuming the following relations:

i) $D(j) \subseteq \Delta, \quad j = 1,\ldots,m$

ii) $D(j) \cap D(i) = \varnothing, \quad \forall i \neq j$

iii) $n = \sum_{j=1}^{m} Card\{D(j)\} \leq Card\{\Delta\}$

For instance, Fig. 6 a), referred to the example of Fig. 5, shows the sets Δ, *D(1)*, *D(2)* and *D(3)*, and shows that VPC1, 2 and 3 are eligible to steal the VPC0 time-slot since $D(j) \subseteq \Delta$, *j* =1, 2, 3. Assuming that the VPC1 and 2 have higher priority with respect to VPC3, the time-slot is stolen by VPC1, since *Card*{*D(1)*} > *Card*{*D(2)*}. Then, assuming that the GES has another available uplink time-slots, Δ is updated as shown in Fig. 6 b); by repeating the procedure, given that VPC2 and 3 are eligible, the time-slot is assigned to VPC2, which has higher priority with respect to VPC3. Finally, even if the GES has another free time-slot, it cannot be used by any VPC because *D(1)*, *D(2)* and *D(3)* are not sub-sets of the updated Δ, as shown in Fig. 6 c).



a) Δ = {1, 2, 3, 4}
D(1) = {1, 2}
D(2) = {3}
D(3) = {3, 4}

Time-slot assigned to VPC1

b) Δ'= Δ \ D(1) = {3,4}

D(2) = {3}
D(3) = {3, 4}

Time-slot assigned to VPC2

c) Δ''= Δ'\ D(2) = {4}

D(3) = {3, 4}

Time-slot not assigned

Fig. 6. Downlink sets for the stolen slot procedure

5 OPNET Simulations

In order to validate the proposed architecture, simulations have been performed with the OPNET tool by MIL3, which is a discrete event simulator specifically developed for simulating telecommunication networks.

The simulation scenario consists of a single gateway transmitting 8 one-to-one VCCs towards the satellite via a TDM uplink. In order to evaluate the performances of the proposed architecture, it is compared to the reference one, in which the VCCs are mapped onto the uplink frame after the connection set-up. The simulation parameters are summarised in Table 1.

Table 1. Simulation parameters

Uplink Capacity	N. of Time-Slots per Frame	Frame Length	Number of RT Connections	Number of RT QoS Classes
9,088,000 bps	1136	53 ms	8	3

QoS Class	PCR Range [bps]	CDV Range [ms]	Application Example
1	16,000 – 1,400,000	10 – 100	Voice
2	64,000 – 40,000,000	100 – 600	Video
3	9,600,00 – 1,500,000	> 600	Files

VCC	PCR [bps]	CDV [ms]	QoS Class	N. of Required Time-Slots per Frame
0	600,000	15	1	75
1	1,300,000	1000	3	162.5
2	64,900	600	2	8.1125
3	10,000	1000	3	1.25
4	1,400,000	100	1	175
5	31,000	10	1	3.875
6	5,000,000	1000	3	625
7	682,100	600	2	85.2625

Four simulation runs have been executed, as shown in Table 2.

Table 2. Simulation runs

Simulation Run	Scenario	Source Rate[1]	VCC Admission Order
1	Reference	Nominal	VCC 0, 1, 2, 3, 4, 5, 6
2	Reference	Nominal	VCC 0, 1, 2, 3, 4, 6, 5
3	Proposed	Nominal	Not significant
4	Proposed	Variable	Not significant

Note that, in the reference scenario, the last VCC that has to be mapped is VCC7; since the other VCCs requires a total amount of 1053 time-slots in order to be mapped onto the uplink frame (as a matter of fact, each VCC is mapped onto an integer num-

[1] The sources can transmit constantly at the PCR – nominal rate – or with a variable rate; the rate variations have a Gaussian distribution with the mean equal to PCR (however, the UPC blocks limit the buffer input rate to the PCR).

ber of time-slots, e.g., VCC1 is mapped onto 163 time-slots), the leftover 83 time-slots are not sufficient to map VCC7. On the contrary, the proposed architecture is capable of accepting the last VCC, since it is not dependent on the frame structure; the sum of the PCRs of the VCCs is equal to the uplink capacity.

Fig. 7 shows the simulation results.

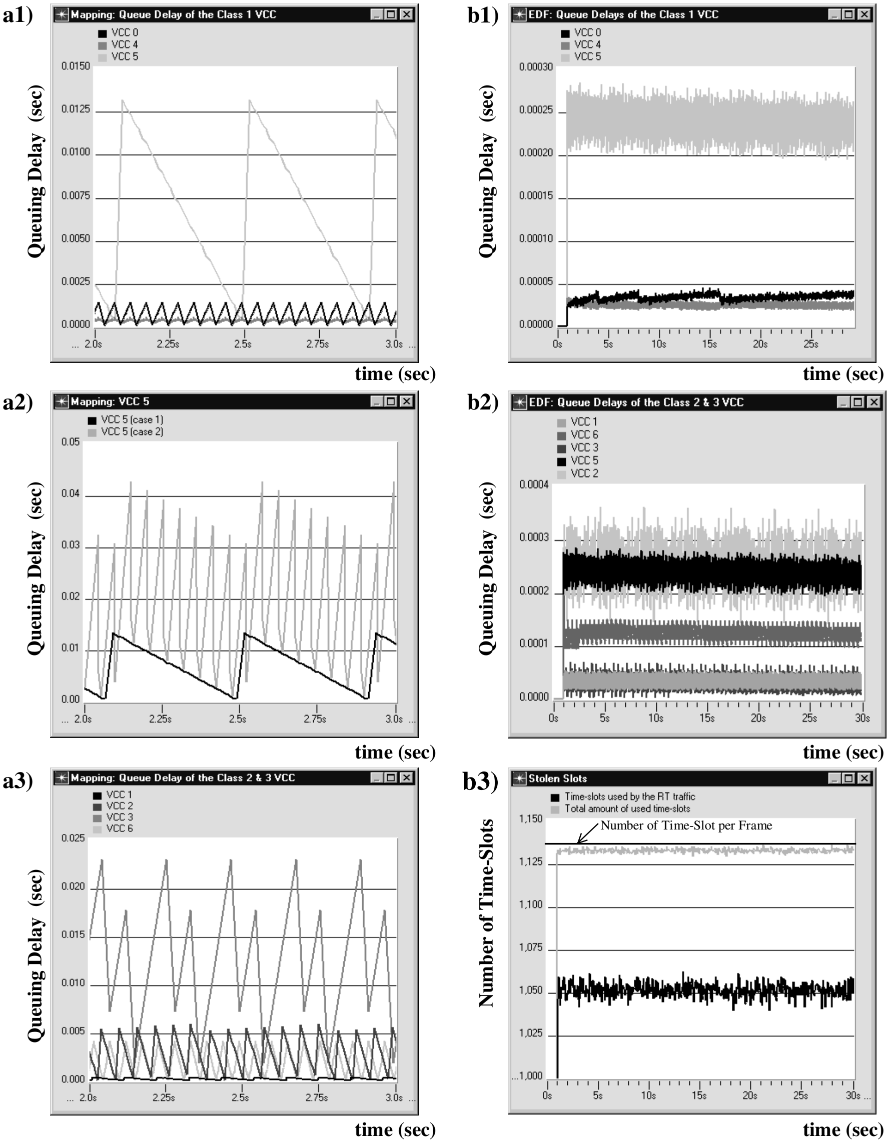


Fig. 7. Simulation results

Fig. 7 a1) shows the queuing delay perceived by the packets of the Class 1 VCCs in Simulation 1. As explained in Section 2, assuming that the time-slots are regularly paced, the queuing delay depends on the PCR of the connection and not on the requested CDV. For instance, the light gray plot shows the queuing delay of VCC5, whose PCR is equal to 31,000 bps; thus, the expected maximum CDV is given by T_{FRAME} / N_{PCR} = 53 ms / 4 = 13.25 ms. The simulation confirms this prediction, and shows that the reference architecture cannot guarantee the requested CDV to VCC5. The periodic saw-tooth behaviour of the queuing delays depend on the fact that the sources are not synchronized with the periodicity of the assigned time-slots.

Fig. 7 a2) shows the queuing delay perceived by the packets of the Class 2 and 3 VCCs in Simulation 1. Also in this case, the queuing delays depend only on the PCR of the connections.

Fig. 7 a3) shows the comparison between the queuing delays perceived by the packets of VCC5 in Simulation 1 and 2. Since in Simulation 2 VCC5 is the last admitted connection, it has to be mapped onto an already heavily loaded uplink frame; thus, the time-slot allocation cannot be as regular as in the Simulation 1, and the maximum distance between two consecutive time-slots is greater. Thus, as shown in Fig. 7 a3), the CDV in Simulation 2 is even greater than the one in Simulation 1 (which was already unacceptable).

Fig. 7 b1) and b2) shows the queuing delay perceived by the packets of the VCCs in Simulation 3. In this case, the queuing delays are independent of the PCR and are much below the requested CDVs, even if the rate of the traffic entering the GES is equal to the uplink capacity – in Simulation 3, the gateway is supporting all of the 8 VCCs transmitting at full rate.

Finally, Fig. 7 b3) shows stolen slot algorithm performances. In Simulation 4, in which the sources transmit with variable rates, the GES transmits a NRT packet every time the stolen slot algorithm communicates that a time-slot is available (see Section 4). The lower plot represents the total transmission rate of the RT VCCs of Simulation 4, while the higher plot represents the total amount of transmitted packets, given by the RT traffic plus the NRT packets transmitted on the stolen slots; the simulation results show that the stolen slot algorithm allows the utilisation of more than 99.8% of the link capacity (note that, in the simulation, a NRT time-slot is available every time a time-slot is stolen).

6 Conclusions

In the present paper, a gateway architecture suitable for DVB-RCS satellite networks has been proposed. The architecture aims at supporting the RT priority traffic in the most efficient manner, without affecting the NRT traffic. At the same time, the scalability issue, which is relevant for the gateways because of the large number of supported traffic flows, is taken into account.

The three objectives are met:

i) The proposed scheduling scheme transmits the RT packets with strict priority with respect to the NRT traffic; the NRT traffic is protected by misbehaving RT sources (i.e., sources transmitting with a higher rate with respect to the contracted one) by the UPC functional blocks of the gateway, which have shaping and policing functionalities.

ii) Thanks to the 'stolen slot' concept, by exploiting the UPC blocks, the proposed 'stolen slot' algorithm is capable of re-distributing the unused capacity which was assigned to RT traffic among the NRT flows.

iii) The proposed two-levels aggregation policy, based on the definition of QoS classes within the RT priority traffic, reduces effectively the number of buffer the gateway has to manage.

Furthermore, the paper defines a procedure that allows the distribution of the available time-slots among unicast and multicast traffic flows.

Finally, simulations have been performed with the OPNET tool, which verified the effectiveness of the proposed buffering and scheduling schemes and of the proposed 'stolen-slot' algorithm.

Acknowledgements. The authors wish to thank Dr. Benoit Garnier of Alcatel Space Industries (F) for his valuable contribution.

References

[1] Pietrabissa, S. Fiorido, "*Access Layer Protocols for the GEOCAST Project*", IST Mobile Communication Summit 2001, Sitges, September 2001

[2] European Broadcasting Union: "*Digital Video Broadcasting (DVB): Interaction channel for satellite distribution systems*", ETSI EN 301 790 V1.2.2, http://www.etsi.org, December 2000

[3] European Broadcasting Union: "*Digital Video Broadcasting (DVB);Interaction channel for Satellite Distribution Systems;Guidelines for the use of EN 301 790*", ETSI TR 101 790 V1.1.1, http://www.etsi.org, September 2001

[4] Garnier, "*Access Layer Specification for GEOCAST System*", GEOCAST Draft, October 2001

[5] ATM Forum Technical Committee: "*Traffic Management Specification Version 4.1*", www.atmforum.org, March 1999

[6] F. Delli Priscoli, A. Faggiano, V. Verrillo: "*Uplink Access Technique in an ATM-based Satellite Network*", Proc. of the 4th ACTS Mobile Communication Summit '99, Sorrento (Italy), June 1999.

[7] H. Koraitim, S. Tohme, H. Cakil, "*MB-ICBT protocol performance in star-configured VSAT satellite networks*", 2nd IEEE Symposium on Computers and Communications (ISCC '97), July 1997

[8] T. Örs, Z. Sun and B.G. Evans, "*A MAC Protocol for ATM over Satellite*", Proceedings of Sixth IEE Conference on Telecommunications, pp. 185-190, Edinburgh-UK, 29 March-1 April 1998

[9] Hung, M.-J. Monpetit, and G. Kesidis, "*ATM via satellite: a framework and implementation*", ACM/ Baltzer WINET, 4(2):141-153, February 1998

[10] V. Firoiu, M. Borden:"A Study of Active Queue Management for Congestion Control", IEEE INFOCOM '99

[11] Chengzhi Li and Edward W. Knightly, "Schedulability Criterion and Performance Analysis of Coordinated Schedulers", in Proceedings of ITC-17, September 2001

Connection Admission Control CAC and Differentiated Resources Allocation RA in a Low Earth Orbit LEO Satellite Constellation

Rima Abi Fadel[1,2] and Samir Tohmé[2]

[1]Ecole Supérieure d'Ingénieurs de Beyrouth
Mkalles, Mar Roukoz, BP 11514, Liban
{rima.abifadel@fi.usj.edu.lb}
[2]Ecole Nationale Supérieure des Télécommunications
46, Rue Barrault, 75013, Paris, France
{abifadel,tohme@ inf.enst.fr}

Abstract. The Up/Down Link UDL of a Low Earth Orbit LEO satellite constellation is a scarce radio resource that needs to be shared efficiently between many users with different needs. A suitable Connection Admission Control CAC policy is required. In our study we assume that the network handles three types of calls: real time (voice) calls with strict constraints over the delay and the bandwidth, non real time (data) calls delay tolerant but with bandwidth guarantees requirements and Best Effort calls with no guarantees requirements. In order to ensure priorities are respected, we use an "enhanced trunk reservation policy" in order to derive the Resources Allocation RA. A differentiated RA scheme is proposed, associated with queuing for the lower priority calls. Different unity bandwidths are associated with calls depending on their requirements. The analytical markovian model is first derived, then the differentiated RA model is compared with two cases of bandwidth granularity choice. Impact of non markovian laws is studied using simulation.

Keywords: Access Networks, Performance Analysis, Satellite Communications.

1 Introduction

LEO Satellite Constellations are seen as a suitable mean for providing mobile users a global access service to terrestrial networks with the main advantage of a small propagation delay compared to the geographically stationary satellites. The general CAC problem within a multi-service LEO Satellite Constellation usually with Inter-Satellite Link ISL is very complex. It is common to split this problem into two components: the CAC associated with the Medium Access Layer MAC at the air interface level and the CAC associated with the establishment of the path within the LEO core network involving the ISL Routing problem.

In this paper we will consider only the CAC problem at the air interface level to the satellite constellation network. This case is found in two typical situations. The first corresponds to a constellation without Inter-Satellite Link ISL like SKYBRIDGE and the second is found in the case of integration of the satellite within the Universal

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 814-825, 2002.

Mobile Telecommunications Service UMTS Satellite Radio Access Network USRAN [4]. A CAC policy based on an "enhanced trunk reservation technique" is used.

The classical trunk reservation technique and a number of its variants have been proposed and widely used in both contexts of routing and handoff. In the routing context, this technique was first applied to telephone networks with a non hierarchical circuit switched environment. By reserving a certain number of trunks for direct-routed traffic in a group, it has been seen as an effective way of stabilizing the network, and preventing performance degradation under overload [8]. In [10], it has been shown that a dynamic trunk reservation policy with a level depending on the traffic load yields better overload performance than a fixed trunk reservation scheme. Optimizing the network performance by a suitable choice of the trunk reservation parameters was studied in [12]. The approach was generalized from the classical telephony context to a multi-service network. In [9], blocking probabilities of multiple traffic streams due to trunk reservation in circuit-switched networks employing adaptive routing were studied. The technique was applied to elastic flows in [5], in a broadband network [6]. A dynamic control has also been shown to outperform the fixed one in a broadband call control admission context in [3]. Other variants also attribute a probability to the acceptance of call once the threshold is reached [13]. In the mobile networks context, the trunk reservation technique, also known as the guard channels technique, is also very popular. It has been used in classical voice cellular networks, as well as in ATM based mobile networks. It is seen as an appropriate way to prioritize calls experiencing handoff over newly generated calls in a cell [7]. Although the utility of using guard channels has been discussed, especially regarding the overall system utilization, and non-reserving schemes were proposed [15,16], the guard channels approach remains popular in the mobile context. In order to increase the system performance, the number of guard channels is varied dynamically using the information concerning ongoing calls in neighboring cells [17] with the mobility pattern [18,19,20]. In [14] the classical guard channels scheme is associated with queuing of originated calls in order to increase channel utilization. A fractional policy is proposed in [21]; it reserves a non integral part of guard channels for hand-off calls by rejecting new calls with some probability that depends on the current channel occupancy. A guard channel scheme is combined with queuing in an integrated voice/data wireless network in [22]. Priority is given to voice handoffs over data handoffs.

In [1], we propose an analytical solution for the "enhanced trunk reservation technique". It consists in admitting a lower priority service until a given threshold is reached, so far identical to the classical scheme, but it admits the blocked service back into the system only when the total number of occupied resources falls back to another threshold. This hysteresis introduction has two main advantages. First, oscillations of the system to and from the blocking state for the lower priority service are excluded. And since, in the satellites context, whenever a transition from a state to another occurs, the Network Control Center NCC, placed on the terrestrial part of the system is to be informed, this leads to less signaling traffic on the air interface.

The scheme proposed is applied to a network with an integration of three service classes. The highest priority class is the real-time service, carrying mainly voice over the allocated channels. Data traffic is separated into two priority classes. The higher priority service has guaranteed bandwidth obtained by reservation that requires a CAC mechanism. The lowest priority data service called “Best Effort” service has no guarantees requirements.

In this paper, the trunk reservation technique is combined with a bandwidth reservation related to the type of call being served.

The paper is organized as follows. In section 2, we introduce the traffic models and parameters that have been used for both the analytical solution and the simulation. The control admission scheme is explained in section 3. In section 4 the analytical solution for the simplified differentiated RA model including queuing for the two data classes is presented. In section 5 this model is compared via simulation to two fixed choices of unity bandwidth allocation per call. Section 6 discusses blocking probabilities experienced by different services and delay experienced by data calls with guarantees when different queue lengths are considered for the differentiated RA scheme together with an enhanced trunk reservation mechanism. In section 7 non markovian service times are chosen and their impact on the above parameters is evaluated. Section 8 concludes the study.

2 Traffic Description

According to the French Réseau National de Recherche en Télécommunications RNRT Satellites Constellation Project [2], a mobile communication access service has been introduced for mobile users using a LEO satellite constellation. The total bandwidth serving pedestrian users on the uplink is of 72 kbits/s, with a granularity of 2.4 kbits/s. The cellular system that will be considered in this paper is based on cells fixed to the earth (similar to the Skybridge LEO Constellation case). Furthermore, the user mobility will be considered as negligible in comparison with the satellite mobility because the typical diameter of a satellite cell is of order of few hundred kilometers and because the only users considered are the pedestrian ones. Thus, we will not consider in our CAC model the issues associated with the Handover HO. Because the bandwidth on the downlink is usually higher than on the up link for pedestrian users, we will focus on the channel allocation on the up link. We assume also that any mobile user will always be able to see at least two satellites at the same time. Three call classes are considered. Voice calls are generated according to a Poisson process with an exponentially distributed call duration time of mean 120s. Channel allocation for voice is considered on the basis of the call level; the period of the ON time in an ON/OFF model being too small to justify the burst mode for voice. Data calls however can be served as bursts. Two types of non real time data calls will be considered. One corresponds to WEB applications, the other to FTP/SMTP data transfer. In all cases, we consider an allocation’s duration based on a request/response transaction scheme. In this paper, we consider different unity bandwidth’s allocation depending on the type of call being served. That is, a bandwidth of 2.4 kbits/s is to be allocated to a voice call entering the system while a 4.8 kbits/s bandwidth is to be allocated to each data or best effort call. This differentiated RA model will be compared to the all 2.4 kbits/s granularity as proposed by the french RNRT project [2] and to another proposition of

4.8 kbits/s granularity for all users. For the WEB applications, an exponentially distributed inter-arrival transactional time - usually known as thinking time - of mean 2 minutes or, equivalently, an arrival rate of $8.33*10^{-3}$ transaction arrivals/second for a given user is assumed. Assuming that a WEB page is represented by a load of 8000 bytes downloaded on a link of 2.4 kbits/s (respectively 4.8 kbits/s) as stated above, the average service time is thus equal to 26.67s (respectively 13.33s). The propagation delay is function of the satellite position covering the user and will be the same no matter what the bandwidth choice.

3 The Connection Admission Control Scheme

The lower priority data calls are accepted into the system until the number of allocated channels reaches a certain threshold s_1. The (higher priority) real-time calls are accepted until all channels are occupied. The data service will be admitted back into the system only when the total number of occupied channels falls back to s_2. Identically, Best Effort is admitted into the system until the number of occupied resources reaches s_3, and is only be admitted back when the total number of occupied resources falls back to s_4. Two separate queues are associated with respectively data calls with guarantees and best effort calls. Calls refused immediate service can still wait in queue until a certain number of channels becomes available.

The problem can be viewed as a generalization of a Birth-Death system with a number of channels equal to N, (30 or 15 channels depending on granularity), (λ_h,μ_h), (λ_l,μ_l) and (λ_{be},μ_{be}) the arrival rates and the inverse of the mean service times for respectively voice, data and best effort services.

Two main configurations for the respective positions of s_2 and s_3 are proposed.

The choice of $s_2 < s_3$, a prudent admission policy, will penalize the data services, but will be much more helpful for the high priority real time voice service. Once blocked, the admission back into the system of the data calls is delayed. Choosing $s_2>s_3$ on the other hand, represents a rather tolerant policy for data calls. Of course transition from one configuration to the other can be done through threshold s_2 variation. Simulation will thus represent different performance measures variations as functions of varying s_2.

4 Exact Analytical Solution for the Simplified Differentiated RA Model

As explained above, we consider that a voice call arriving to the system will be allocated a 2.4 kbits/s bandwidth while a data call, or a best effort call, is to be served with a 4.8 kbits/s bandwidth. This means that if an arriving lower priority call, when its admission into the system is allowed, finds less available bandwidth than its 4.8 kbits/s requirement, it will be queued if the queue is not full. The non differential bandwidth allocation scheme has been analyzed, via simulation, in [11].

In order to simplify the analytical model derivation, thresholds s_1 and s_2 on one hand and s_3 and s_4 on the other, are assumed to be identical.

The process consists in a five dimensional, continuous time, discrete states Markov process. The solution is given under stationary conditions (when time tends to infinity). Although the equations, given in the appendix, are written for the case for which data and best effort calls require a double bandwidth than the one needed for voice calls, the approach remains the same when other granularities are chosen.

Figure 1.a) shows the evolution of the blocking experienced by data calls and voice calls as threshold s_1 varies, for different values of the data queue length. Best effort queue length is fixed to a value of 5. The blocking experienced by best effort calls is not represented because in the configuration studied, these calls represent a proportion of 10% of the total system load while voice calls are responsible for 50% and data calls for the remaining 40%.

Two important observations can be made. As s_1 increases, the blocking experienced by data calls decreases while the blocking probability for voice increases. Of course, increasing s_1 means that less channels are reserved for the exclusive use of voice. On the other hand, for a given value of s_1, as the data queue length increases, data calls experience less blocking. In fact, when a queue exists, a data call is not completely rejected once threshold s_1 is reached. It will be allowed to wait until a sufficient number of resources becomes available. The longer the queue is, the more calls are admitted to wait, and the less blocking situations will occur. A data call is rejected if, upon its arrival into the system, it cannot directly access the system and the queue is full. However, the longer the queue is, the longer a call will have to wait. The delay aspect will be considered in the simulation.

Figure 1.b) represents the impact of the choice of threshold s_3 over the blocking probabilities of data and best effort calls. Numerical values show a very light impact over the blocking of voice calls, the latter is thus not represented here. This is due to the light load associated with best effort. The data queue length is fixed to a value of 5, and the best effort queue length is varied. The same discussion concerning the impact of data queue length over data calls can be made to explain the impact of the best effort queue length over best effort calls. Figure 1.b) also shows a very important sensibility of best effort blocking probability over s_3.

5 Granularity Impact over Blocking Probabilities and Advantages of the Differentiated RA Scheme

Simulation uses the QNAP simulation environment. The program was first validated for special cases for which the exact solution is known. The parameters used for the simulation are described in section 2 above. Voice calls, data calls and best effort calls are assumed to be respectively responsible for 50%, 40% and 10% of the traffic carried by each of the systems considered.

Both the tolerant and the prudent policy are studied by varying threshold s_2. The transition from a configuration to the other is made for $s_2 = s_3$. The other thresholds are fixed to the values of s_1=87%, s_3=53%, s_4=27% of the total number of available channels. The percentages are deduced from the necessity of having integer values for these thresholds. We first consider a system without queuing.

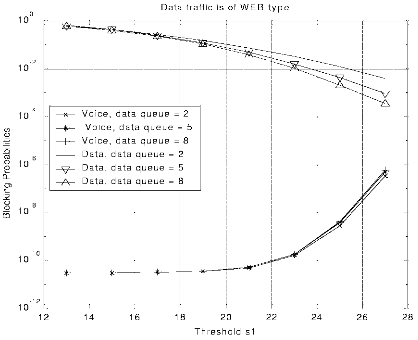


Fig. 1. a) Blocking probabilities for voice and data as functions of threshold s_1.

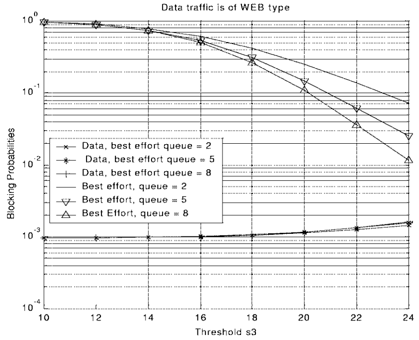


Fig. 1. b) Blocking probabilities for data and best effort as functions of threshold s_3.

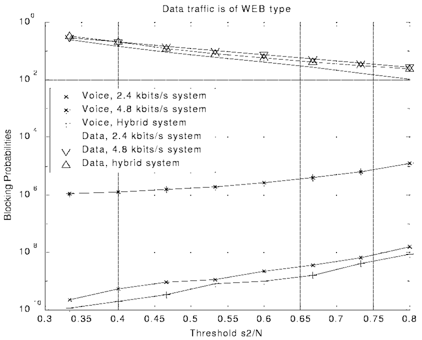


Fig. 2. Blocking probabilities for voice and data calls for two different granularity cases and for the differentiated RA scheme.

Figure 2 shows that data service blocking probability is very sensitive to the choice of s_2, decreasing as s_2 increases, and thus resulting in an increase in the voice calls blocking probability. The situation corresponds to a case where data calls are, once blocked, admitted earlier back into the system. It can also be seen that for a prudent policy, voice calls experience considerable improvement in their blocking performance. This was the basic idea for which the prudent policy was thought of.

The best effort blocking performance is not represented. Numerical values however show a slight increase of the best effort blocking probability with s_2. In fact, as s_2 increases, more data calls can be admitted into the system, both the two other service classes will be affected.

Figure 2 also shows that, when the two systems at 2.4 kbits/s and 4.8 kbits/s fixed granularities are equally loaded, the smaller granularity system, although carrying a higher traffic, presents better blocking performance regarding voice calls and data calls. This is due to the fact that the overall number of available circuits is bigger when granularity is smaller. The problem with the 4.8 kbits/s system is also that, although data calls are quicker served, and thus leave the system earlier, voice calls will occupy the unit resource for the same amount of time as in the 2.4 kbits/s system, while occupying twice as much bandwidth. This situation naturally leads to more blocking.

The whole idea of introducing a differentiated RA scheme is to gain the advantages of both systems. Data calls, for the same amount of data to be transferred, once admitted into the system, will finish service and leave sooner, as in the 4.8 kbits/s granularity system, but without the drawback of wasting bandwidth over voice calls that are still served at a smaller unit bandwidth. We consider the same load factor for the differentiated RA scheme. Figure 2 shows that regarding voice calls, the blocking performance is even better than what was observed for the 2.4 kbits/s case. This is explained by the fact that data calls will leave the system sooner than for the 2.4 kbits/s case, leaving behind twice as much idle resources as in the 2.4 kbits/s system.

Regarding data calls however, the 2.4 kbits/s system still shows the best blocking situation. One must not forget that for a given value of the blocking threshold of data calls in the differentiated RA scheme, the total number of occupied resources must be at least two unities smaller than s_1 for the data call to be accepted. Otherwise, the available bandwidth is not sufficient to serve data. Admission of data calls is thus subject to more constraints in the differentiated RA scheme than in the 2.4 kbits/s system. The all 4.8 kbits/s system shows the most important blocking probability for the same relative value of the threshold s_2, due mainly to the smaller number of available circuits for this system.

6 Impact of the Data Queue Length on Different Performance Measures

In this section, only the differentiated RA scheme is considered. Blocking probabilities of different traffic classes are evaluated as functions of the blocking out threshold of data calls s_2, for different values of the data queue length. The best effort queue length is fixed to a value of 5. The simulation is run for the fixed values of $s_1 = 26$, $s_3 = 16$ and $s_4 = 8$, under the same relative contributions of the different services to the total traffic as in the previous section. Simulation results are given in figure 3.

Figure 3.a) represents the evolution of voice and data blocking probabilities with threshold s_2 and the data queue length. For a tolerant policy and a given $s_2 > s_3 = 16$, increasing the data queue length will be clearly advantageous for the blocking probability of data calls. In fact, when a data call does not have the possibility to be immediately served, under the queuing hypothesis, it will wait in queue until the admission conditions into the system are fulfilled. The call is rejected when the queue is saturated, a less frequent situation when the queue length increases. Of course, this means that voice calls will be penalized.

For a prudent policy, the advantage of queuing is less perceptible. The three curves representing the blocking experienced by data calls on one hand, and the three ones representing the blocking of voice on the other tend to converge for a very strict admission policy. When s_2 is very small, blocking out data calls will be considerably delayed. No matter what the queue length, it will very quickly be saturated and arriving calls will quickly start to be rejected anyway. The advantage of having more room for queuing is not considerable when severe blocking constraints are applied to data calls.

Figure 3.b) shows the waiting time in queue of data calls evolution as a function of the threshold s_2 and for different values of the queue length. It can be seen that for a given s_2, the bigger the queue length, the higher the delay. This is even more visible for prudent policies with small values of s_2. This is the drawback of having less blocking situations for the data calls. Another observation is the very important delay decrease when s_2 increases. This is due to the fact that, as s_2 increases, data calls are blocked over a smaller interval of the occupied resources. This means that, after s_1 is reached, threshold s_2 will be reached much faster when it increases, and so it will be possible to serve the clients in queue that in turn, experience less waiting time.

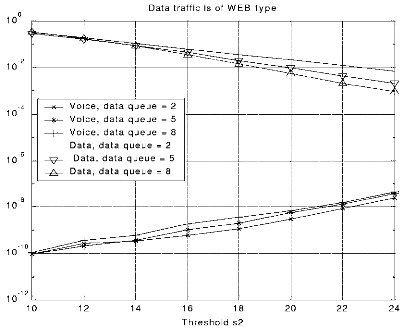


Fig. 3. a) Blocking probabilities of voice and data calls.

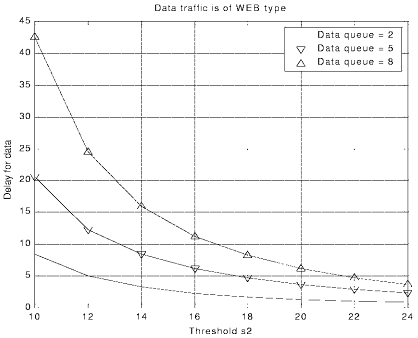


Fig. 3. b) Delay for data calls in seconds.

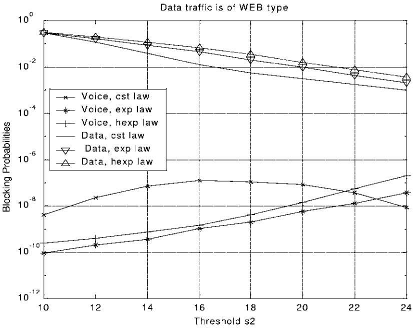


Fig. 4. a) Blocking probabilities of voice and data calls.

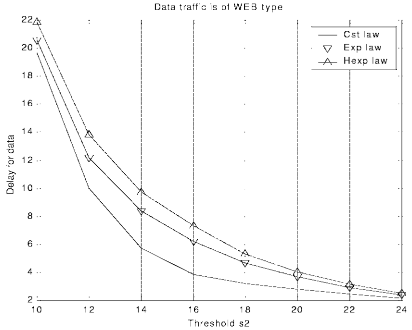


Fig. 4. b) Delay for data calls in seconds.

An important conclusion is that if a strict control policy is chosen for data admission control, increasing the queue length will be more costly, will induce a very high waiting time, without much improvement in the blocking performance of data.

Due to the very strict admission control associated with best effort, its blocking probability is high and almost insensitive at all to the choice of s_2. The waiting time in the queue follows the severe blocking conditions with a high value that can attain an order of few minutes.

7 Impact of Different Service Time Distributions for the Data Classes on Different Performance Measures

In this section, the differentiated RA scheme blocking performance as well as the delay of data calls are evaluated for different distributions of the service time of the two lower priority data services. The service time for voice calls remains however exponentially distributed.

Two distributions are considered together with the exponential hypothesis of the previous sections. All distributions are assumed to have the same mean. We consider constant, exponential and hyper-exponential distributions respectively characterized with a variation coefficient of 0,1 and a chosen value of 2.

These laws are analyzed in order to see if the conclusions of the previous section still hold when more general, and thus realistic hypothesis are considered. One conclusion to be discussed is the impact of the choice of s_2 over the blocking probabilities of voice and data calls and over the waiting time for data. The previous section has shown that a good choice of s_2 should be done in the perspective of a compromise between the blocking probabilities of voice and data and the delay of data. The latter decreases when s_2 increases. Simulation results are given in figure 4.

The conclusion made for the delay still hold for more general laws. Increasing s_2 appears to be advantageous for the three cases of the variation coefficient being considered. This can be seen in figure 4.b). Another observation is the natural increase of the delay, for a given s_2, with the variation coefficient. The best delay is observed for the constant law, with a null variance.

Regarding the blocking performance for voice and data calls, we can see that the conclusion concerning the choice of s_2, still holds when the two lower priority services are served with a hyper-exponential distribution for the service time. When this service time is constant however, figure 4.a) shows that the choice of a very tolerant policy, $s_2 \approx s_1$, slightly penalizes voice calls compared to a very strict control policy for data calls, while considerably improving the blocking of data calls. An appropriate decision in this case would be to choose a very tolerant policy for data.

8 Conclusion

In this paper an enhanced trunk reservation mechanism associated with a differentiated RA scheme has been studied.

Simulation has shown that, compared to systems with fixed bandwidth allocation of 2.4 kbits/s and 4.8 kbits/s respectively for each call, no matter what the class of call, the differentiated scheme offers the best blocking performance for the high priority voice calls. Another advantage is that the faster transmission of data, provided by the 4.8 kbits/s system, is still possible with the differentiated RA scheme.

Queuing has proven to be more advantageous regarding the blocking of data calls. Waiting time certainly increases with the buffer size. An increase that is no longer justified when data is subject to severe blocking constraints. The improvement of the blocking probability when increasing the buffer size is not of importance for this case.

A suitable dimensioning of the buffer size is however a must when a tolerant admission policy scheme is used.

Finally, simulation has also shown that s_2 has to be chosen in a way as to have a suitable blocking probability for data calls without much penalizing voice: s_2 is to have an intermediate value between the very tolerant and the very strict admission control. However, numerical values show that even for very tolerant policies, blocking probability for voice remains considerably small. On the other hand, the choice of a tolerant policy is more suitable when data and best effort bursts have constant lengths. This means that in all cases, a tolerant policy with a high value of s_2, appears to be a fair scheme. The impact of such a decision over the signaling traffic must however be evaluated in order to determine the exact values of the threshold parameters.

References

[1] Rima Abi Fadel, Samir Tohmé. *Connection Admission Control CAC and Resources Allocation RA on the Up/Down Link UDL in a Low Earth Orbit LEO Satellite Constellation.* Proc. IEEE Int. Symp. Comp. and Commun., Hammamet, Tunisia, Jul.2001.

[2] Poethi Boedhihartono, Gérard Maral. *Contribution to the Description of Mission Scenario for Handover Studies.* ENST June 2000 – RNRT Satellite Constellation Project. WP 2.2 on Handover and Associated Signalling

[3] Heba Koraitim, Samir Tohmé. *Resource Allocation and Connection Admission Control in satellite Networks*, IEEE J. Select. Areas Commun., Vol. 17, No. 2, Feb. 1999.

[4] H.Koraitim, G.Schäfer, S.Tohmé. *Quality of Service Aspects of Transport Technologies for the UMTS Radio Access Network.* IFIP Personal Wireless Communication Conference PWC'2000, Published by Kluwer, September 2000, Gdansk, Poland.

[5] Sara Oueslati. *QOS Routing of Elastic Flows in Multi-service Networks*, PhD Dissertation, Ecole Nationale Superieure des Telecomm. Paris, Nov. 2000.

[6] P.Tran-Gia, F.Hubner. *An Analysis of Trunk Reservation and Grade of Service Balancing Mechanisms in Multiservice Broadband Networks.* IFIP TC6 Modelling and Performance Evaluation Workshop. La Martinique, 1993.

[7] B.Tripathi, A.Kumar. *Performance analysis of microcellisation with channel reservation for supporting two mobility classes in cellular wireless networks.* Proc. IEEE Int. Conf. on Pers. Wireless Commun. ICPWC, India, 1997.

[8] R.S. Krupp. *Stabilization of alternate routing network.* IEEE Int. Comm. Conference, Philadelphia, PA, 1982.

[9] M. Rajaratnam, F. Takawira. *Modelling Multiple traffic streams in Circuit-Switched Networks.* IEEE Global Telecomm. Conf. GLOBECOM '96. 'Commun.: The Key to Global Prosperity , Vol. 1.

[10] Ren P. Liu, Peter J. Moylan. *Dynamic Trunk Reservation for teletraffic links.* IEEE Proc, Global Telecomm. Conf., GLOBECOM '95., Vol. 1.

[11] Rima Abi Fadel, Samir Tohmé. *Hybrid Connection Admission Control CAC in a Low Earth Orbit LEO Satellite Constellation.* IFIP Workshop on IP and ATM Traffic Management, WATM'01, Paris, September 2001.

[12] V. Anantharam, M. Benchekroun. *Trunk Reservation based control of circuit switched networks with dynamic routing.* IEEE Proc. of the 29th Conf. on Decision and Control. Honolulu, Hawaii. Dec. 1990.

[13] T. Oda, Y. Watanabe. *Optimal Trunk Reservation for a Group with Multislot Traffic Streams.* IEEE Trans. Commun., Vol. 38, no. 7, July 1990.

[14] Roch Guérin. *Queuing - Blocking System with Two Arrival Streams and Guard Channels.* IEEE Trans. on Commun., Vol. 36, no. 2, Feb. 1998.

[15] C.H. Yoon, C.K. Un. *Performance of personal portable radio telephone systems with and without guard channels.* IEEE J. Select. Areas Commun., vol. 11, issue 6 , Aug. 1993.

[16] B. Narendran, P. Agrawal, D.K. Anvekar. *Minimizing Cellular Handover Failures without Channel Utilization Loss.* Proc. IEEE Global Telecomm. Conf., 1994. GLOBECOM '94.

[17] O.T.W. Yu, V.C.M. Leung. *Self-tuning prioritized call handling mechanism with dynamic guard channels for mobile cellular systems.* Proc. IEEE 46th Vehic. Technol. Conf., 1996. Mobile Technology for the Human Race, vol. 3.

[18] A. L. Beylot, S. Boumerdassi, G. Pujolle. *A new prioritized handoff strategy using channel reservation in wireless PCN.* Proc. IEEE Global Telecomm. Conf., GLOBECOM 1998. The Bridge to Global Integration, vol. 3.

[19] O.T.W Yu, V.C.M. Leung. *Adaptive resource allocation for prioritized call admission over an ATM-based wireless PCN.* IEEE J. Select. Areas Commun., Sept. 1997, vol. 15, iss. 7.

[20] M.H. Chiu, M.A. Bassiouni. *Predictive schemes for handoff prioritization in cellular networks based on mobile positioning.* J. Select. Areas Commun., Mar. 2000, vol. 18, iss. 3.

[21] R. Ramjee, R. Nagarajan, D. Towsley. *On optimal call admission control in cellular networks.* Proc. IEEE INFOCOM '96, vol.1, pp. 43 -50.

[22] D. Calin. *A probabilistic model for handling voice and data traffic in wireless networks.* IEEE Int. Conf. on Universal Personal Commun., 1998. ICUPC '98, vol. 1.

Appendix

i, j and k are respectively the number of voice calls, data calls and best effort calls being served. l and m are respectively the number of data calls and best effort calls being queued. N represents the total number of available channels. supl and supm denote respectively the capacity of the data queue and the best effort queue. $1_{\text{condition}}$ is a variable set to 1 when the condition holds true, and is equal to 0 otherwise.

Equations of the analytical solution, numerically under C++, are the following.

For k varying from 0 to supk = int(s_3/2)
For j varying from 0 to supj = min(int(s_1/2),int((N-2*k)/2))
For i varying from 0 to supi = (N-2*j-2*k):

If (i+2*j+2*k) $\le s_3 - 2$:

$$\begin{aligned}(\lambda_h+\lambda_l+\lambda_{be}+i\mu_h+j\mu_l+k\mu_{be})\pi_{i,j,k,0,0} &= 1_{i>0}\lambda_h\pi_{i-1,j,k,0,0}+1_{j>0}\lambda_l\pi_{i,j-1,k,0,0}\\ &+1_{k>0}\lambda_{be}\pi_{i,j,k-1,0,0}+1_{i<\sup i}(i+1)\mu_h\pi_{i+1,j,k,0,0}+1_{j<\sup j}(j+1)\mu_l\pi_{i,j+1,k,0,0}\\ &+1_{k<\sup k}(k+1)\mu_{be}\pi_{i,j,k+1,0,0}\end{aligned} \tag{1}$$

If (i+2*j+2*k) = s_3 –1 or (i+2*j+2*k) = s_3:

$$\begin{aligned}&(\lambda_h+\lambda_l+1_{m<\mathrm{sup}\,m}\lambda_{be}+i\mu_h+j\mu_l+k\mu_{be})\pi_{i,j,k,0,m}=1_{m=0}1_{i>0}\lambda_h\pi_{i-1,j,k,0,m}\\&+1_{m>0}1_{i>0}1_{(i+2j+2k)=s_3}\lambda_h\pi_{i-1,j,k,0,m}+1_{m=0}1_{j>0}\lambda_l\pi_{i,j-1,k,0,m}+1_{m=0}1_{k>0}\lambda_{be}\pi_{i,j,k-1,0,m}\\&+1_{i<\mathrm{sup}\,i}(i+1)\mu_h\pi_{i+1,j,k,0,m}+1_{k>0}1_{(i+2j+2k)=s_3}1_{m<\mathrm{sup}\,m}1_{i<\mathrm{sup}\,i}(i+1)\mu_h\pi_{i+1,j,k-1,0,m+1}\\&+1_{j<\mathrm{sup}\,j}(j+1)\mu_l\pi_{i,j+1,k,0,m}+1_{k>0}1_{(i+2j+2k)>s_3-2}1_{m<\mathrm{sup}\,m}1_{j<\mathrm{sup}\,j}(j+1)\mu_l\pi_{i,j+1,k-1,0,m+1}\\&+1_{m>0}\lambda_{be}\pi_{i,j,k,0,m-1}+1_{k<\mathrm{sup}\,k}(k+1)\mu_{be}\pi_{i,j,k+1,0,m}+1_{m<\mathrm{sup}\,m}k\mu_{be}\pi_{i,j,k,0,m+1}\end{aligned}\tag{2}$$

If s3 < (i+2*j+2*k) ≤ s_l –2:

$$\begin{aligned}&(\lambda_h+\lambda_1+1_{m<\mathrm{sup}\,m}\lambda_{be}+i\mu_h+j\mu_1+k\mu_{be})\pi_{i,j,k,0,m}=1_{i>0}\lambda_h\pi_{i-1,j,k,0,m}\\&+1_{j>0}\lambda_1\pi_{i,j-1,k,0,m}+1_{m>0}\lambda_{be}\pi_{i,j,k,0,m-1}+1_{i<\mathrm{sup}\,i}(i+1)\mu_h\pi_{i+1,j,k,0,m}\\&+1_{j<\mathrm{sup}\,j}(j+1)\mu_1\pi_{i,j+1,k,0,m}+1_{k<\mathrm{sup}\,k}(k+1)\mu_{be}\pi_{i,j,k+1,0,m}\end{aligned}\tag{3}$$

If (i+2*j+2*k) = s_l –1 or (i+2*j+2*k) = s_l:

$$\begin{aligned}&(\lambda_h+1_{l<\mathrm{sup}\,l}\lambda_l+1_{m<\mathrm{sup}\,m}\lambda_{be}+i\mu_h+j\mu_l+k\mu_{be})\pi_{i,j,k,l,m}=\\&1_{l=0}1_{i>0}\lambda_h\pi_{i-1,j,k,l,m}+1_{l>0}1_{i>0}1_{(i+2j+2k)=s_l}\lambda_h\pi_{i-1,j,k,l,m}\\&+1_{l=0}1_{j>0}1_{(i+2j+2k)>s_3}\lambda_l\pi_{i,j-1,k,l,m}+1_{l>0}\lambda_l\pi_{i,j,k,l-1,m}+1_{m>0}\lambda_{be}\pi_{i,j,k,l,m-1}\\&+1_{i<\mathrm{sup}\,i}(i+1)\mu_h\pi_{i+1,j,k,l,m}+1_{j>0}1_{(i+2j+2k)=s_l}1_{l<\mathrm{sup}\,l}1_{i<\mathrm{sup}\,i}(i+1)\mu_h\pi_{i+1,j-1,k,l+1,m}\\&+1_{j<\mathrm{sup}\,j}(j+1)\mu_l\pi_{i,j+1,k,l,m}+1_{l<\mathrm{sup}\,l}1_{(i+2j+2k)>s_l-2}\,j\mu_l\pi_{i,j,k,l+1,m}\\&+1_{k<\mathrm{sup}\,k}(k+1)\mu_{be}\pi_{i,j,k+1,l,m}\\&+1_{j>0}1_{k<\mathrm{sup}\,k}1_{l<\mathrm{sup}\,l}1_{(i+2j+2k)>s_l-2}(k+1)\mu_{be}\pi_{i,j-1,k+1,l+1,m}\end{aligned}\tag{4}$$

If s_l < (i+2*j+2*k) < N:

$$\begin{aligned}&(\lambda_h+1_{l<\mathrm{sup}\,l}\lambda_1+1_{m<\mathrm{sup}\,m}\lambda_{be}+i\mu_h+j\mu_1+k\mu_{be})\pi_{i,j,k,l,m}=\\&1_{i>0}\lambda_h\pi_{i-1,j,k,l,m}+1_{l>0}\lambda_1\pi_{i,j,k,l-1,m}+1_{m>0}\lambda_{be}\pi_{i,j,k,l,m-1}\\&+1_{i<\mathrm{sup}\,i}(i+1)\mu_h\pi_{i+1,j,k,l,m}+1_{j<\mathrm{sup}\,j}1_{(i+2j+2k)<N-1}(j+1)\mu_1\pi_{i,j+1,k,l,m}\\&+1_{k<\mathrm{sup}\,k}1_{(i+2j+2k)<N-1}(k+1)\mu_{be}\pi_{i,j,k+1,l,m}\end{aligned}\tag{5}$$

For (i+2*j+2*k) = N:

$$\begin{aligned}&(1_{l<\mathrm{sup}\,l}\lambda_1+1_{m<\mathrm{sup}\,m}\lambda_{be}+i\mu_h+j\mu_1+k\mu_{be})\pi_{i,j,k,l,m}=\\&1_{i>0}\lambda_h\pi_{i-1,j,k,l,m}+1_{l>0}\lambda_1\pi_{i,j,k,l-1,m}+1_{m>0}\lambda_{be}\pi_{i,j,k,l,m-1}\end{aligned}\tag{6}$$

The equation written for i=N, j=k=l=m=0 is replaced by the equation stating that the sum of all state probabilities of the system is equal to one, in order to ensure that the numerical problem has only one solution.

Dimensioning Bandwidth for Elastic Traffic

Zhong Fan

Marconi Labs Cambridge
William Gates Building, JJ Thomson Avenue
Cambridge CB3 0FD, UK
zhong.fan@marconi.com

Abstract. In this paper, we discuss the issue of dimensioning Internet access lines for elastic traffic. This is important for Internet service providers (ISPs) because over-dimensioning wastes precious bandwidth resources, while under-dimensioning generally leads to less satisfactory quality of service (QoS) perceived by subscribers. Our discussion is based on the M/G/R processor sharing model which characterizes TCP traffic at flow level. Our analysis demonstrates the impact of a number of key factors (and their relations) on the dimensioning procedure. We consider two dimensioning methods based on different QoS criteria. It is found that the method based on the delay factor is superior in that both the average delay (throughput) and blocking performance targets can be satisfied. Numerical and theoretical analyses also illustrate that significant multiplexing gain can be achieved for elastic flows and this gain increases with burstiness.

1 Introduction

It has been recognized that there are generally two classes of traffic in the current Internet, namely, stream traffic and elastic traffic [1]. Typical stream services are real-time video and voice services, while elastic services could be file transfers, emails, web pages and other data traffic based on TCP. At the moment a large amount of Internet traffic is elastic and therefore it is essential for ISPs to dimension Internet access lines properly to cater for the service needs of elastic traffic. Dimensioning should also allow for statistical multiplexing to achieve better utilization of network resources.

Over the past few years there have been extensive studies on IP traffic characterization, in particular, long-range dependence and self-similarity (see e.g., [2]). While many papers have focused on packet level behavior, recently a number of studies show that processor sharing (PS) models provide a simple and accurate characterization of elastic IP traffic at flow level [3] [4] [5] [6]. Nabe et al. [3] use an M/G/1 PS model to discuss a design methodology of the Internet access network as well as document caching at a proxy server. A drawback of the M/G/1 PS model is that it assumes that one TCP connection is able to utilize the total link capacity by its own, which is not true in reality. In [5] and [7], PS models have been used to demonstrate the need of admission control for TCP

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 826–837, 2002.

flows. In general, PS models are able to successfully capture the elastic properties of traffic generated by closed loop control transport protocols (e.g., TCP) without going into complicated details of packet level traffic characteristics.

In this paper, we discuss the issue of dimensioning Internet access lines for elastic traffic. This is very important for Internet service providers because over-dimensioning wastes precious bandwidth resources, while under-dimensioning generally leads to less satisfactory quality of service (QoS) perceived by subscribers. Our discussion is based on the M/G/R processor sharing model which characterizes TCP traffic at flow level. The performance criteria of dimensioning could be average transfer delay and throughput, both of which are related to a so-called delay factor. We also consider parameters such as blocking probability and multiplexing gain. Among the two dimensioning methods in which we are interested here, we have found that the method based on the delay factor is superior in that both the average delay (throughput) and blocking performance targets can be satisfied. Both numerical and theoretical analyses illustrate that significant multiplexing gain can be achieved for elastic flows and this gain increases with burstiness.

2 The M/G/R PS Model

Here we consider a simple scenario (shown in Figure 1) where subscribers are connected to an access multiplexer via customer access lines (e.g., ADSL lines) and then to the core network (where servers reside) via an access trunk line. In this context, the trunk line must have enough capacity to accommodate both upstream and downstream traffic loads.

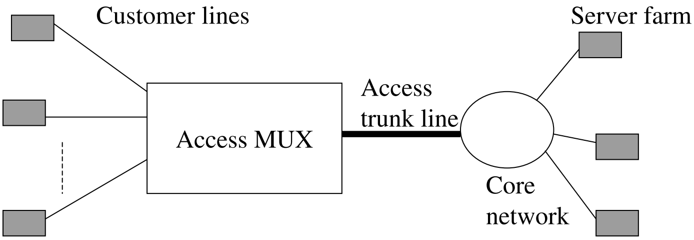


Fig. 1. Access network

We assume that elastic traffic is generated by file transfer applications. The flow (TCP connection) arrival process is Poisson [1] and the file size distribution

[1] The Poissonian assumption is appropriate when the considered link is shared by a very large number of users [5].

has heavy tails, e.g., Pareto distribution. Actually, an important advantage of PS models is that results derived from PS models are insensitive to file size distributions [8]. The file transmission rates are controlled by the TCP feedback algorithm as a function of network congestion. When TCP works ideally, the access trunk line can be modelled as a processor sharing queue. It is known that with the PS scheduling discipline large files do not delay small ones too much when compared with FIFO scheduling [9].

Let r_p and C denote the limited peak rate of an individual subscriber (peak access line rate, e.g., modem speed or the rate limited by the maximum TCP window size) and the trunk line capacity respectively [2]. Then the link appears like a PS system with R servers where R is an integer and $R = C/r_p$, hence the name M/G/R PS queue. If θ is the average file size and λ is the average flow arrival rate, then the traffic load (or utilization) ρ is $\theta\lambda/C$. It has been shown in [9] [10] that the mean conditional sojourn time $T(x)$ for a file of size x is given by

$$T(x) = \frac{x}{r_p}(1 + \frac{E_2(R, R\rho)}{R(1-\rho)}) \tag{1}$$

where E_2 represents Erlang's second formula:

$$E_2(R, R\rho) = \frac{A}{B(1-\rho) + A} \tag{2}$$

where $A = (R\rho)^R/R!, B = \sum_{i=0}^{R-1}(R\rho)^i/i!$.

As in [9], define a delay factor f_R as

$$f_R = 1 + \frac{E_2(R, R\rho)}{R(1-\rho)}. \tag{3}$$

Then $T(x)$ can be re-written as $T(x) = \frac{x}{r_p} f_R$. And the mean throughput γ is given by

$$\gamma = x/T(x) = r_p/f_R. \tag{4}$$

The delay factor f_R represents the increase of the average file transfer time (and decrease of the average throughput) due to link congestion. For the special case of $R = 1$ (M/G/1 PS), $f_R = \frac{1}{1-\rho}$, and $\gamma = C(1-\rho)$. Note that in [8], a similar demerit factor is introduced.

It has been advocated by Roberts et al. [7] that TCP admission control should be implemented so that flows sharing a bottleneck can achieve some minimum throughput. Let r_m denote the minimum fair share which can be used as an admission control threshold. Then the upper limit of the number of admitted flows, N, is C/r_m. In this case, the blocking probability, F, is a function of the parameters N, R and ρ [9]:

$$F(N, R, \rho) = \frac{AD(1-\rho)}{B(1-\rho) + A(1-D\rho)} = \frac{E_2 D(1-\rho)}{1 - E_2 D\rho} \tag{5}$$

[2] Here for simplicity, we assume that all subscribers have the same maximum access rate. For different access rates, we could use an average value of R in dimensioning as suggested by [9].

where A, E_2 and B are as defined previously, and D is given by $D = \rho^{N-R}$. Note that when $R = N$, (5) reduces to Erlang's first formula. When $R = 1$, (5) becomes

$$F(N, \rho) = \rho^N (1 - \rho)/(1 - \rho^{N+1}). \tag{6}$$

Figure 2 shows delay factor f_R as a function of ρ with different R. It can be seen that when load is low, $f_R \approx 1$, therefore $\gamma \approx r_p$. In this case, throughput is nearly full access line rate. However, as load increases, f_R increases dramatically, thus throughput drops sharply. The delay improves while R increases. This can also be seen in Figure 3, in which normalized throughput (γ/C) is shown as a function of ρ. It is easy to see that as ρ is very close to 1, $\gamma \approx C(1 - \rho)$.

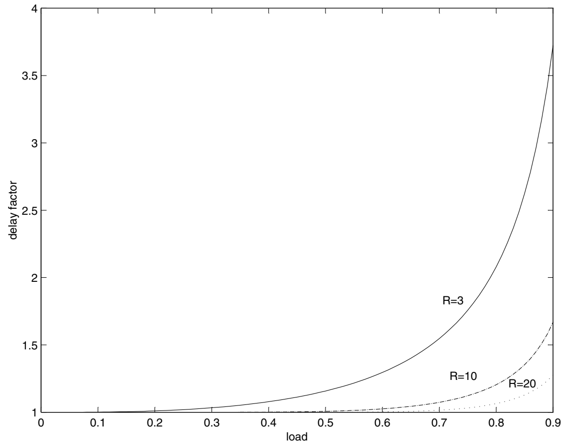


Fig. 2. Delay factor vs. load

Figure 4 shows the relationship between blocking probability and f_R with different N. Here $R = 10$. Since both F and f_R are monotonically increasing functions of ρ, F increases with f_R. We can see for more elastic traffic $(N \gg R)$, the blocking probability is smaller. In terms of dimensioning, we could provision the trunk line so as to have both a small blocking probability (say, 0.001) and a desirable delay factor (say, 1.01).

In [10] the accuracy of the M/G/R PS model is studied using simulations and the basic applicability of this model to access link dimensioning is confirmed. We further their work by considering two different dimensioning procedures and investigating in more depth the impact of the delay factor and other parameters (such as blocking probability) on network performance.

3 Access Trunk Line Dimensioning

One possible dimensioning method uses the blocking probability as a QoS criterion [1]. Similar to the situation in telephony networks, blocking probability

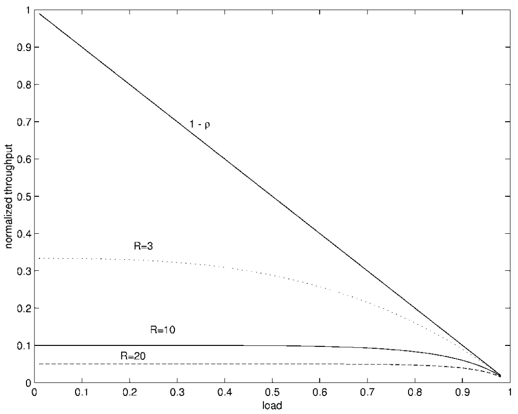


Fig. 3. Normalized throughput vs. load

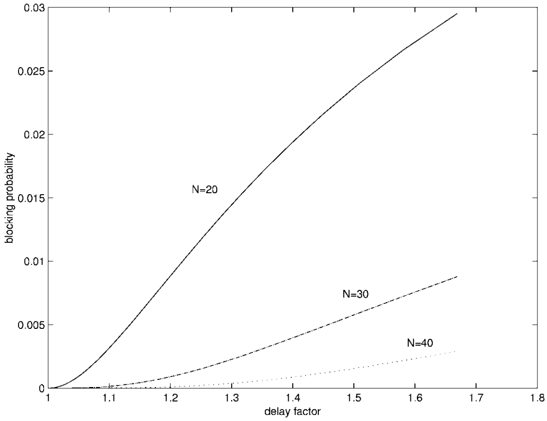


Fig. 4. Blocking probability against delay factor

has a significant impact on user satisfaction. As an alternative, we can also base our dimensioning decision on the delay factor described in the previous section. For elastic Internet services such as file transfer and web traffic, large delay contributes greatly to the user-perceived quality degradation.

3.1 Dimensioning Method One

For link dimensioning purposes, the above model needs to be extended to arbitrary link rates, i.e., R does not have to be an integer [5]. In this case, let R denote the integer part of C/r_p. Then we have

$$T(x) = \frac{x}{r_p} f_R \tag{7}$$

and

$$\gamma = r_p / f_R, \tag{8}$$

where

$$f_R = 1 + \frac{(1 - (\frac{C}{r_p} - R)(1-\rho))r_p E_2(R, C\rho/r_p)}{C(1-\rho)}. \tag{9}$$

To dimension the trunk line for elastic traffic, we use the delay factor f_R as the QoS measure since f_R determines both average transfer time (Eqs. (1) and (7)) and throughput (Eqs. (4)and (8)) for a given flow. Obviously, f_R has to be chosen greater than 1 (but preferably close to 1). For a given f_R, we can numerically solve (9) to obtain the desired capacity value, C.

Figure 5 shows the dimensioning result of the trunk capacity (normalized capacity w.r.t. r_p) for medium to high loads with different target delay factors. As shown in the figure, f_R has significant impact on capacity, especially at high loads. For instance, the required capacity for f_R of 1.2 is roughly double that for f_R of 1.5 at the load of 0.95. Therefore the delay factor is indeed an appropriate QoS measure for elastic traffic. Assume admission control is implemented to ensure that the load is smaller than 1. In this case, the blocking probabilities as function of ρ are shown in Figure 6 for the dimensioning case of $f_R = 1.5$. All the blocking probabilities are very small (close to zero) when $\rho < 0.8$. However, at high loads (say, $\rho = 0.95$), blocking probabilities differ significantly for different N. It can be seen that for a dimensioned capacity satisfying the delay factor criterion, it is possible to achieve a target blocking probability (especially at high loads) by choosing the value of N.

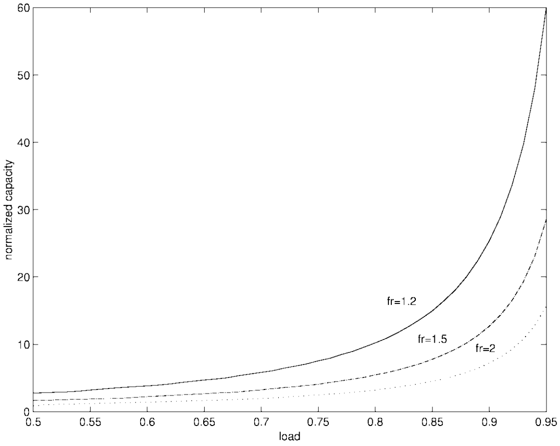


Fig. 5. Trunk line capacity vs. load

As another more realistic example, we use the data traces from [11], which was obtained in an ADSL field trial in Germany in 1998. Some of the parameters

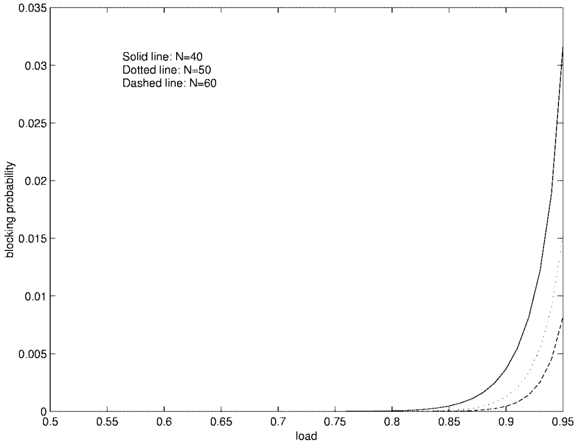


Fig. 6. Blocking probability vs. load when $f_R = 1.5$

observed for HTTP over TCP/IP traffic in active client access sessions are: downstream access line rate $r_p = 2.5$ Mbps, mean rate $m = 10.5$ kbps. Figure 7 shows the dimensioning result for the trunk capacity with $f_R = 1.01$. Also shown is the required capacity calculated based on the sum of the mean rates of all sources.

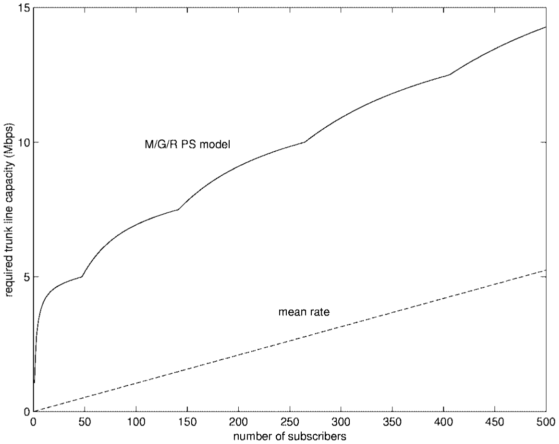


Fig. 7. Trunk line capacity vs. number of subscribers

3.2 Dimensioning Method Two

Figure 8 plots the dimensioning result when we use the blocking probability as the QoS criterion. The target blocking probability is 0.001 and the peak access rate r_p is 2.5 Mbps. It can be seen that as the minimum throughput r_m increases, the required capacity increases. Figure 9 shows the corresponding delay factor f_R under the dimensioned trunk capacities. As expected, f_R improves (decreases) as the minimum throughput r_m increases. For larger r_m, f_R remains more or less the same over a wide range of ρ. However, there are some oscillations for the case of $r_m = 0.5$ Mbps and f_R increases quite significantly as ρ approaches 1. This is because as ρ grows, the required trunk line capacity increases, therefore R increases. Thus, the fact that ρ and R have opposite impact on f_R (as shown in Figure 2) explains the oscillation phenomenon.

From the above discussion, we can see that for the dimensioning based on the blocking probability and minimum throughput, it is difficult to obtain a desirable delay factor. Hence, it can result in unsatisfactory average delay and throughput performance. For example, for $r_m = 0.5$ Mbps, f_R is always above 1.25. In this sense, the dimensioning scheme based on the delay factor is more suitable because the blocking performance can also be tuned therein.

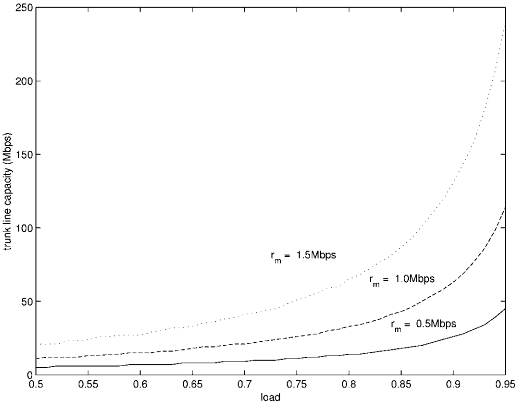


Fig. 8. Trunk line capacity vs. load

4 Multiplexing Gains

Define the multiplexing gain G as

$$G = \frac{nr_p}{C}, \tag{10}$$

where n is the number of sources. The maximum possible value of gain is obtained when dimensioning is based on the mean rate, i.e., $C = nm$. Therefore, $G_{max} =$

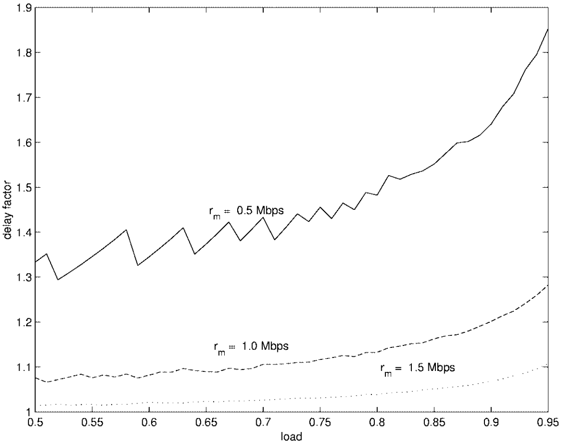


Fig. 9. Delay factor vs. load

r_p/m. Intuitively, this means that for highly bursty traffic, with $m \ll r_p$, G can be very large. However, this maximum gain cannot be attained in reality, because the average bandwidth dimensioning is unacceptable in terms of QoS.

Figure 10 shows G versus n with an f_R of 1.01, using the data in [11]. It can be seen that G increases with n, which demonstrates the benefit of exploiting the statistical features of elastic traffic. In fact, when ρ is close to 1, $r_p/f_R \approx C(1-\rho)$, hence we have

$$C \approx r_p/f_R + nm. \tag{11}$$

So,

$$G \approx \frac{nr_p}{r_p/f_R + nm} = \frac{r_p}{\frac{r_p}{nf_R} + m}. \tag{12}$$

It is clear that as $n \to \infty, G \to G_{max}$. (12) also shows that G increases with f_R.

Next we investigate the impact of the source activity factor $p = m/r_p$ on the multiplexing gain G. Figure 11 shows G as a function of p when n is 100 and f_R is 1.01. G decreases with p. In other words, multiplexing gains increase with burstiness. As a special case, it can be seen in (12) that G is indeed decreasing with p. This is also consistent with the results in [12] where multiplexing gains for stream traffic in an ATM QoS context is considered.

5 Discussion

For small documents, TCP slow start dominates the file transfer phase. In [6], the effect of the slow start phase and round trip times (RTT) are taken into consideration. The transfer time is thus a number of RTTs more than the result predicted by the PS model. However, when the document size is sufficiently

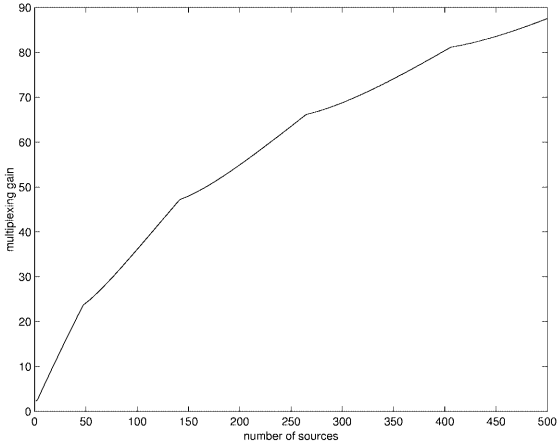


Fig. 10. Multiplexing gain vs. number of sources

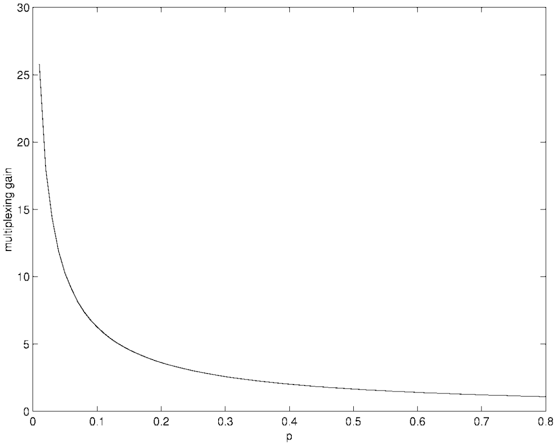


Fig. 11. Multiplexing gain vs. source activity factor

large, the M/G/R PS model is accurate enough for dimensioning purposes [10]. Moreover, since the mean transaction time for small files is small anyway, the inaccuracy of the model does not affect users' perception of quality very much. On the other hand, further refinement of the model to deal with short file transfers is a direction of future research.

In some cases, the TCP window control mechanism is not "ideal". For example, during congestion when packets are lost and retransmissions become necessary, the successful transfers of files proceed at a total rate that can be well below C. In this case, we need to introduce a link efficiency factor $\alpha(\alpha \leq 1)$ such that the average file transfer rate is αC. A discussion on how to estimate α can be found in [8]. From a dimensioning point of view, we can have a safety margin for the link capacity, i.e., set the link capacity C as C'/α, where C' is obtained using the PS model described before.

6 Conclusion

In this paper, we discuss the issue of dimensioning Internet access lines for elastic traffic. Our discussion is based on the M/G/R processor sharing model which characterizes TCP traffic at flow level. Our analysis demonstrates the impact of a number of key factors (and their relations) on the dimensioning procedure. We consider two dimensioning methods based on different QoS criteria. It is found that the method based on the delay factor is superior in that both the average delay (throughput) and blocking performance targets can be satisfied. Both numerical and theoretical analyses illustrate that significant multiplexing gain can be achieved for elastic flows and this gain increases with the number of sources and traffic burstiness.

Acknowledgements. I would like to thank A. Berger of Akamai and D. Heyman of AT&T for their useful communications.

References

1. J. Roberts. Realizing quality of service guarantees in multi-service networks. In *IFIP PMCCN*, 1997.
2. A. Feldmann et al. Data networks as cascades: explaining the multifractal nature of Internet WAN traffic. In *ACM SIGCOMM*, 1998.
3. M. Nabe et al. Analysis and modelling of world wide web traffic for capacity dimensioning of Internet access lines. *Performance Evaluation*, 34:249–271, 1998.
4. A. Berger and Y. Kogan. Dimensioning bandwidth for elastic traffic in high-speed data networks. *IEEE/ACM Trans. Networking*, 8(5):643–654, 2000.
5. S. Ben Fredj et al. Statistical bandwidth sharing: a study of congestion at flow level. In *ACM SIGCOMM*, 2001.
6. J. Beckers et al. Generalized processor sharing performance models for Internet access lines. In *9th IFIP Conference on Performance Modelling and Evaluation of ATM and IP Networks*, 2001.
7. L. Massoulie and J. Roberts. Arguments in favour of admission control for TCP flows. In *ITC 16*, 1999.
8. D. P. Heyman et al. A new method for analysing feedback-based protocols with applications to engineering web traffic over the Internet. In *ACM SIGMETRICS*, 1997.
9. K. Lindberger. Balancing quality of service, pricing and utilization in multiservice networks with stream and elastic traffic. In *ITC 16*, 1999.
10. A. Riedl et al. Investigation of the M/G/R processor sharing model for dimensioning of IP access networks with elastic traffic. In *First Polish-German Teletraffic Symposium*, 2000.

11. J. Charzinski. Fun factor dimensioning for elastic traffic. In *ITC Specialist Seminar on IP Measurement, Modeling and Management*, 2000.
12. Z. Fan and P. Mars. Multiplexing gains in ATM networks. In D. Kouvatsos, editor, *Performance Analysis of ATM Networks*, pages 377–395. Kluwer Academic Publishers, 2000.

Fair Adaptive Bandwidth Allocation: A Rate Control Based Active Queue Management Discipline

Abhinav Kamra, Huzur Saran, Sandeep Sen*, and Rajeev Shorey**

Department of Computer Science and Engineering,
Indian Institute of Technology,
Hauz Khas, New Delhi 110016, India
{saran, ssen, srajeev}@cse.iitd.ernet.in

Abstract. We propose Fair Adaptive Bandwidth Allocation (FABA), a buffer management discipline that ensures a fair bandwidth allocation amongst competing flows even in the presence of non-adaptive traffic. FABA is a rate control based active queue management discipline that provides explicit fairness and can be used to partition bandwidth in proportion to pre-assigned weights. FABA is well-suited for allocation of bandwidth to aggregate flows as required in the differentiated services framework. We study and compare FABA with other well known queue management disciplines and show that FABA ensures fair allocation of bandwidth across a much wider range of buffer sizes at a bottleneck router. FABA uses randomization and has an O(1) average time complexity, and, is therefore scalable. The space complexity of the proposed algorithm is O(B) where B is the buffer size at the bottleneck router. We argue that though FABA maintains per active-flow state, through O(1) computation, reasonably scalable implementations can be deployed which is sufficient for network edges and ISPs.

1 Introduction

Active Queue Management (AQM) disciplines [4] are needed at intermediate routers since they provide protection for adaptive traffic from aggressive sources that try to consume more than their "fair" share, These schemes ensure "fairness" in bandwidth sharing and provide early congestion notification.

A typical Internet gateway is characterized by multiple incoming links, a single outgoing link of bandwidth C packets per second and a buffer size of B packets. The queue management discipline, operating at the enqueuing end, determines which packets are to be enqueued in the buffer and which are to be dropped. The scheduling discipline, at the dequeuing end, determines the order in which packets in the buffer are to be dequeued. Combinations of queue

* Supported in part by an IBM UPP award.

** Research Staff Member, IBM India Research Laboratory, New Delhi, Adjunct Faculty CSE Department, I.I.T, New Delhi.

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 838–849, 2002.

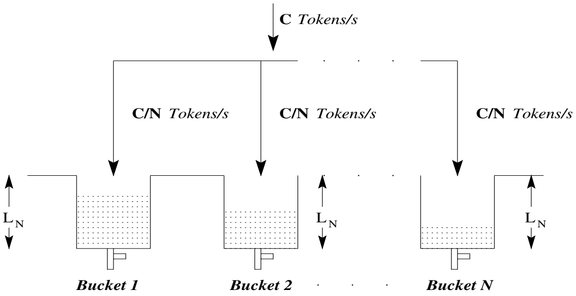


Fig. 1. FABA architecture for rate control using token buckets

management and scheduling disciplines can provide early congestion detection and notification and can be used in the differentiated services framework.

In this paper we present Fair Adaptive Bandwidth Allocation (FABA), a queue management discipline which when coupled with even the simplest scheduling discipline, such as First Come First Served (FCFS), achieves the following goals: (i) fair bandwidth allocation amongst flows, aggregates of flows and hierarchy, (ii) congestion avoidance by early detection and notification, (iii) low implementation complexity, (iv) easy extension to provide differentiated services.

In a recent work [6], the authors proposed Selective Fair Early Detection (SFED) algorithm. SFED is a rate control based buffer management algorithm. For the sake of completeness, we describe SFED in detail in this paper. FABA, proposed and described in this paper is an extension of SFED.

FABA is an active queue management algorithm that deals with both adaptive and non-adaptive traffic while providing incentive for flows to incorporate end-to-end congestion control. It uses a rate control based mechanism to achieve fairness amongst flows. Further, congestion is detected early and notified to the source. FABA is easy to implement in hardware and is optimized to give $O(1)$ complexity for both enqueue and dequeue operations. We compare FABA with other queue management schemes such as RED [1], CHOKe [3], Flow Random Early Drop (FRED) [2], and observe that FABA performs at least as well as FRED and significantly better than RED and CHOKe. However, when buffer sizes are constrained, it performs significantly better than FRED.

The paper is organized as follows. An overview of FABA is presented in Section 2. Section 2.1 explains our proposed algorithm in detail. We compare the performance of FABA with other buffer management algorithms namely RED, CHOKe and FRED with the help of simulations conducted using the ns2 [5] network simulator in Section 3. We summarize our results and discuss future work in Section 4.

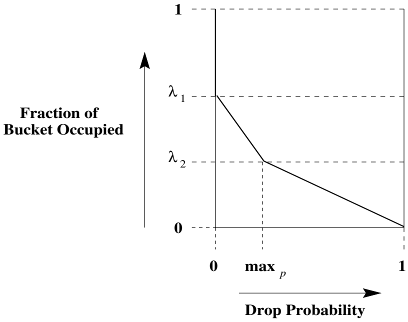


Fig. 2. Probability profile of dropping a packet for a token bucket

2 An Overview of FABA

Since FABA is an extension of SFED algorithm [6], for the sake of completeness, we begin with a description of SFED algorithm in this section. Selective Fair Early Detection (SFED) is an easy to implement rate control based active queue management discipline which can be coupled with any scheduling discipline. SFED operates by maintaining a token bucket for every flow (or aggregate of flows) as shown in Figure 1. The token filling rates are in proportion to the permitted bandwidths. Whenever a packet is enqueued, tokens are removed from the corresponding bucket. The decision to enqueue or drop a packet of any flow depends on the occupancy of its bucket at that time. The dependence of drop probability on the buffer occupancy is shown in Figure 2.

SFED ensures early detection and congestion notification to the adaptive source. A sending rate higher than the permitted bandwidth results in a low bucket occupancy and so a larger drop probability thus indicating the onset of congestion at the gateway. This enables the adaptive flow to attain a steady state and prevents it from getting penalized severely. However, non-adaptive flows will continue to send data at the same rate and thus suffer more losses.

Keeping token buckets for flows is different from maintaining an account of per-flow queue occupancy. The rate at which tokens are removed from the bucket of a flow is equal to the rate of incoming packets of that flow, but the rate of addition of tokens in a bucket depends on its permitted share of bandwidth and not on the rate at which packets of that particular flow are dequeued. In this way a token bucket serves as a control on the bandwidth consumed by a flow. Another purpose of a token bucket is to keep a record of the bandwidth used by its corresponding flow in the recent past. This is important for protocols such as TCP which leave the link idle and then send a burst in alternate periods. No packet of such a flow should be dropped if its arrival rate averaged over a certain time interval is less than the permitted rate. The height of the bucket thus represents how large a burst of a flow can be accommodated. As such, this scheme does not penalize bursty flows unnecessarily.

For the simplest case, all buckets have equal weights and each token represents a packet[1] (see Figure 1), the rate at which tokens are filled in a bucket is given by $R_N = C/N$ tokens per second, where C is the outgoing link capacity in packets per second and N is the number of *active* flows. We say that a flow is active as long as its corresponding bucket is not full. Note that this notion of active flows is different from that in the literature [4,2,1], namely, a flow is considered active when the buffer in the bottleneck router has at least one packet from that flow. However, we will see later, that despite the apparent difference in the definitions of an active flow, the two notions are effectively similar.

A flow is identified as inactive whenever while adding tokens to that bucket, it is found to be full. The corresponding bucket is then immediately removed from the system. Its token addition rate is compensated by increasing the token addition rate for all other buckets fairly. Similarly, the tokens of the deleted buckets are redistributed among other buckets.

The heights of all the buckets are equal and their sum is proportional to the buffer size at the gateway. Since the total height of all the buckets is conserved, during the addition of a bucket the height of each bucket decreases, and during the deletion of a bucket the height of each bucket increases. This is justified since if there are more flows a lesser burst of each flow should be allowed while if there are lesser number of flows a larger burst of each should be permitted. During creation and deletion of buckets, the total number of tokens is conserved. This is essential since if excess tokens are created a greater number of packets will be permitted into the buffer resulting in a high queue occupancy while if tokens are removed from the system a lesser number of packets will be enqueued which may result in lesser bursts being accommodated and also lower link utilization in some cases.

With reference to Figure 1 and Figure 2, we define the system constants and variables below.

Constants: B is the size of the buffer (in packets) at the gateway, α is the parameter that determines the total number of tokens, T, in the system. We assume $\alpha = 1$ throughout the paper and $T = \alpha B$, $\alpha > 0$. $\lambda_1, \lambda_2, \mathrm{max}_p$ are constants between 0 and 1 and are illustrated in Figure 2. The probability profile f_p maps the bucket occupancy to the probability of dropping a packet when it arrives.

A probability profile is needed to detect congestion early and to notify the source by causing the packet to be dropped or marked. Figure 2 shows the probability profile used in our simulations. It is similar to the gentle variant of RED drop probability profile (Figure 2).

$$f_p(x_i) = \begin{cases} 0 & \lambda_1 < \frac{x_i}{L_N} < 1 \\ max_p\left(\frac{\lambda_1 - x_i/L_N}{\lambda_1 - \lambda_2}\right) & \lambda_2 < \frac{x_i}{L_N} < \lambda_1 \\ max_p + & \\ (1 - max_p)\left(\frac{\lambda_2 - x_i/L_N}{\lambda_2}\right) & \\ & 0 < \frac{x_i}{L_N} < \lambda_2 \end{cases}$$

[1] For simplicity, we make the assumption that all packets have the same size

Global variables: N is the number of flows (equal to the number of buckets) in the system. L_N is the maximum height of each bucket when there are N active connections in the system, $L_N = T/N$. A variable σ keeps account of the excess or deficit of tokens caused at deletion and creation of buckets. During bucket addition and deletion, the total number of tokens in the system may temporarily deviate from its constant value T. σ is used to keep track of and compensate for these deviations.
Per-flow variables: x_j is the occupancy of the *ith* bucket in tokens, $0 \leq x_j \leq L_N$.

2.1 The SFED Algorithm

The events that trigger actions at the gateway are (i) arrival of a packet at the gateway, (ii) departure of a packet from the gateway.
Arrival of a packet (flowid j): If the bucket j does not exist, we create bucket j. A packet is dropped with probability $p = f_p(x_j)$. If the packet is not dropped, $x_j = x_j - 1$, and the packet is enqueued.
Departure of packet: The model shown in Figure 1 is valid only for fluid flows where the flow of tokens into the buckets is continuous. However, the real network being discrete due to the presence of packets, the model may be approximated by adding tokens into the system on every packet departure since the packet dequeue rate is also equal to C. When there are no packets in the system, i.e., every bucket is full, no tokens will be added into the system as required. The steps taken on each packet departure are: (i) $\sigma = \sigma + 1$, (ii) if $(\sigma > 0)$ distribute(σ). [2]
Token distribution: Tokens may be distributed in any manner to ensure fairness. One straightforward way is to distribute them in a round robin fashion as follows. As long as $(\sigma \geq 1)$, we find next bucket j, $x_j = x_j + 1$, $\sigma = \sigma - 1$, and if bucket j is full, we delete bucket j.
Creation of bucket j: A bucket is created when the first packet of a previously inactive flow is enqueued thus increasing the flows to $N + 1$. The rate of filling tokens into each bucket is $R_{N+1} = C/(N+1)$. A full bucket is allocated to every new flow. It is ensured that the total number of tokens in the system remain constant. The tokens required by the new bucket are compensated for by the excess tokens generated, if any, from the buckets that were shortened. Variable σ is maintained to compensate for the excess or deficit of tokens. The following are the steps: (i) Increment active connections to $(N + 1)$, (ii) Update height of each bucket to $L_{N+1} = T/(N + 1)$, (iii) $x_j = T/(N + 1)$, (iv) $\sigma = \sigma - L_{N+1}$, (v) For every bucket $i \neq j$, if $(x_i > L_{N+1})$, $\sigma = \sigma + (x_i - L_{N+1})$, $x_i = L_{N+1}$.
Deletion of bucket j: A bucket is deleted when a flow is identified to be inactive thus reducing the number of flows to $N - 1$. The rate of filling tokens into each bucket is increased to $R_{N-1} = C/(N - 1)$. It is obvious that the deleted bucket is full at the time of deletion. Variable σ is maintained to compensate for the excess tokens created so that the total tokens remain constant. The steps are:

[2] FABA attains higher efficiency by using randomization for this token distribution step

(i) decrement active connections to $(N-1)$, (ii) update height of each bucket to $L_{N-1} = T/(N-1)$, (iii) $\sigma = \sigma + L_N$.
Remark: Note that by the nature of token distribution, if a packet is in position x from the front of the queue, the average number of tokens added to its bucket when the packet is dequeued is $\frac{x}{N}$. Therefore, when the bucket is full (and deleted), there is no packet in the buffer belonging to that flow. This is consistent with the existing definition of active flows.

We see that token distribution and bucket creation are $O(N)$ steps. Hence, in the worst case, both enqueue and dequeue are $O(N)$ operations. We now present the FABA Algorithm which is O(1) for both enqueue and dequeue operations. This extension makes the FABA algorithm scalable, and hence, practical to implement as compared to the SFED algorithm.

2.2 The FABA Algorithm

We now propose the FABA algorithm that has O(1) average time complexity for both enqueue and dequeue operations.
Arrival of a packet (flowid j): If the bucket j does not exist, create bucket j. If $x_j > L_N$ (to gather excess tokens), $\sigma += x_j - L_N$, $x_j = L_N$. Drop the packet with probability $p = f_p(x_j)$. If the packet is not dropped, $x_j = x_j - 1$, enqueue packet in the queue.
Departure of packet: The steps taken on a packet departure are (i) $\sigma = \sigma + 1$, (ii) Let $\beta = \max\left(1, \frac{\sigma}{Q+1}\right)$, where Q is the queue size after the packet departure, (iii) $distribute(\beta)$.
Token distribution: $\beta = \frac{\sigma}{Q+1}$ buckets are accessed randomly. Now instead of always adding tokens, we try to keep the number of spare tokens (σ) as close to 0 as possible. This means if we are short of tokens, i.e., $\sigma < 0$, then we grab tokens from the buckets, else we add tokens. Any bucket that has more than L_N tokens is deleted. From a number between 1 to β (including 1 and β), choose a random bucket j. If $\sigma > 0$, $x_j = x_j + 1$ and $\sigma = \sigma - 1$, else, $x_j = x_j - 1$, $\sigma = \sigma + 1$. If $x_j > L_N$, we delete bucket j.

Note that σ may be negative but it is balanced from both sides, i.e., when it is negative we try to increase it by grabbing tokens from the buckets and when it is positive we add tokens to the buckets. The quantity β is the upper bound on the work done in the token distribution phase. But generally, and as also observed in our experiments, β is always close to 1.

Creation of bucket j: The steps associated with this are (i) increment active connections to $(N+1)$, (ii) update $L_{N+1} = T/(N+1)$, (iii) $x_j = T/(N+1)$, (iv) $\sigma = \sigma - L_{N+1}$.

In FABA, we do not gather the excess tokens that might result in the buckets when we decrease the height. Hence, this procedure becomes O(1). Instead these excess tokens are removed whenever the bucket is accessed next.

Deletion of bucket j: The steps associated with bucket deletion are (i) decrement active connections to $(N-1)$, (ii) update height of each bucket to $L_{N-1} = T/(N-1)$, (iii) $\sigma = \sigma + x_j$.

Every operation in the FABA algorithm is O(1) in the amortized sense. This can be seen from the following observation. If K packets have been enqueued till now, then, the maximum number of tokens added to the buckets over successive dequeues is also K, implying $O(1)$ amortized cost for token distribution. All other operations such as bucket creation, deletion, etc, are constant time operations.

If the Buffer size of the bottleneck router is B, then it is easy to see that the space complexity of FABA algorithm is O(B). This can be argued as follows: the number of leaky buckets is equal to the number of active flows passing through the router and in the worst case, there are B active flows at the bottleneck buffer. We have already argued that by the nature of token distribution, if a packet is in position x from the front of the queue, the average number of tokens added to its bucket when the packet is dequeued is $\frac{x}{N}$. Therefore, when the bucket is active, it is highly likely that there is at least one packet in the buffer belonging to that flow.

3 Simulation Results

We compare the performance of FABA with other active queue management algorithms in different network scenarios. RED and CHOKe are O(1) space algorithms and make use of the current status of the queue only to decide the acceptance of an arriving packet, whereas FRED keeps information corresponding to each flow. This makes FRED essentially O(N) space. FABA also keeps one variable per active-flow. This extra information per active-flow is made use of to provide better fairness. All simulations are done using Network Simulator (ns) [5]. We use FTP over TCP NewReno to model adaptive traffic and a Constant Bit Rate (CBR) source to model non-adaptive traffic. The packet size throughout the simulations is taken to be 512 Bytes. For RED and FRED, *min_th* and *max_th* are taken to be 1/4 and 1/2 of the buffer size, respectively. The value of max_p is 0.02 for RED, FRED and FABA. The values of λ_1 and λ_2 are 1/2 and 1/4 for FABA. All values chosen for the algorithms correspond to those recommended by the respective authors.
Time Complexity of the Proposed Algorithm: The time complexity of FABA has two parts, namely creation or deletion of buckets and the number of tokens distributed to the buckets for every packet departure. The first two operations take constant time but the number of tokens distributed (Section 2.2) is $\frac{\sigma}{Q+1}$. Over a sequence of packet departures, the amortized cost of a packet departure is $O(1)$, therefore it may be more appropriate to discuss the worst case for a single distribution step. Unless there are a large number of buckets created or deleted consecutively, the quantity $\frac{\sigma}{Q+1}$ is no more than two (see [7]). We have seen that the average number of tokens distributed is 1 almost everywhere.

3.1 Fair Bandwidth Allocation

The Internet traffic can broadly be classified into adaptive and non-adaptive traffic. An adaptive flow reduces its sending rate in response to indications of

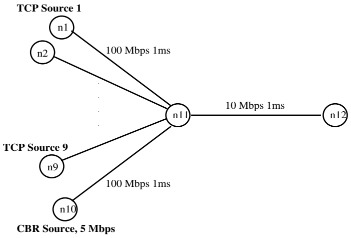


Fig. 3. The Simulation Topology

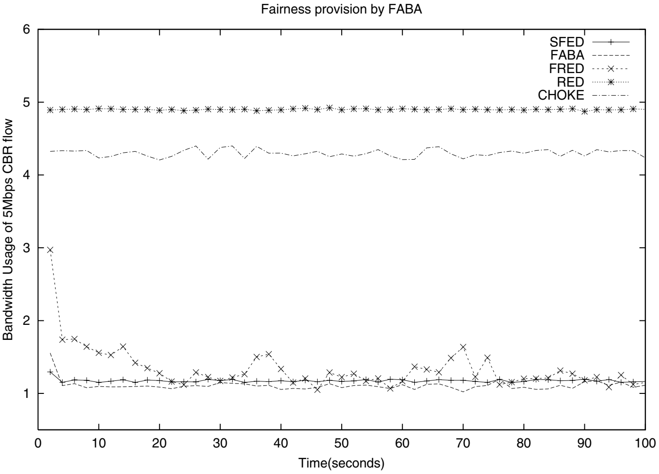


Fig. 4. Bandwidth allocated to a heavy (10 Mbps) CBR flow by different schemes

congestion in its network path. We show the performance of different queue management disciplines in providing fairness when adaptive traffic competes with non-adaptive traffic. The simulation scenario is shown in Figure 3. The bottleneck link capacity is 10 Mbps. A CBR flow sends at 5 Mbps while 9 TCPs share the bottleneck link with the CBR flow. We denote by P_{bd}, the bandwidth delay product of a single TCP connection, which is 78 packets in this example. The buffer size at the bottleneck link is set to $10P_{bd}$ since there are a total of 10 flows competing. In Figure 4, we see how much bandwidth the heavy CBR flow can grab with different buffer management schemes. Since the bottleneck link capacity is 10 Mbps, the fair share of the CBR flow is 1Mbps. However, since CBR is always sending data at a higher rate and the TCP rates are not constant, the CBR flow gets at least its fair share of throughput i.e., 1 Mbps. We observe that FABA performs better in terms of fairness since it does not allow the bulky CBR flow to grab much more than its fair share of the link capacity.

Table 1. Bandwidth with 9 TCP flows at various buffer sizes as a fraction of their fair share

	$\frac{1}{10}P_{total}$	$\frac{1}{5}P_{total}$	$\frac{1}{2}P_{total}$	P_{total}	$2P_{total}$
RED	0.551	0.564	0.559	0.557	0.556
CHOKe	0.622	0.631	0.659	0.685	0.688
FRED	0.814	0.873	0.945	0.961	0.962
SFED	0.923	0.975	0.982	0.990	0.994
FABA	0.888	0.953	0.975	0.987	0.993

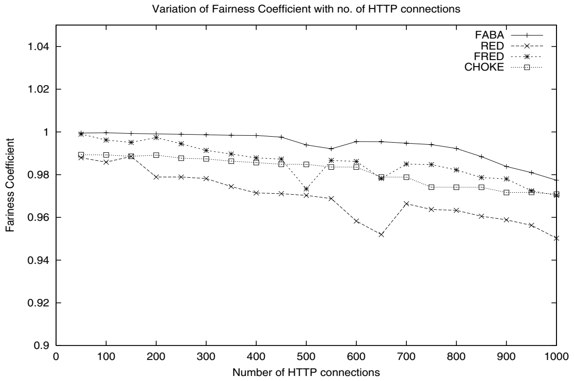


Fig. 5. Fairness coefficient versus number of HTTP connections for different AQM schemes

3.2 Performance with Varying Buffer Sizes

For a buffer management scheme to perform well, there should be enough buffering capacity available at the bottleneck gateway. We now see how well FABA performs with a variation in the buffer space. In the above simulation, the bandwidth delay product of all the flows combined is $P_{total} = 780$ packets. Table I shows the average bandwidth obtained by the 9 TCP flows combined as a fraction of their fair share with varying buffer space at the bottleneck link.

From table I, it is clear that FABA consistently performs better than RED, CHOKe and FRED across a wide range of buffer sizes. further, we observe that FABA performs almost as good as SFED. Since FABA has lower time complexity than SFED, it is only appropriate to study the performance of FABA rather than SFED.

3.3 Performance of FABA with Different Applications

We now study the fairness property of the FABA algorithm. We use the well known definition of fairness index. If f is the fairness index, r_i is the throughput of connection i, and the total number of connections is N, then $f = \frac{\left(\sum_{i=1}^{N} r_i\right)^2}{N\left(\sum_{i=1}^{N} r_i^2\right)}$.

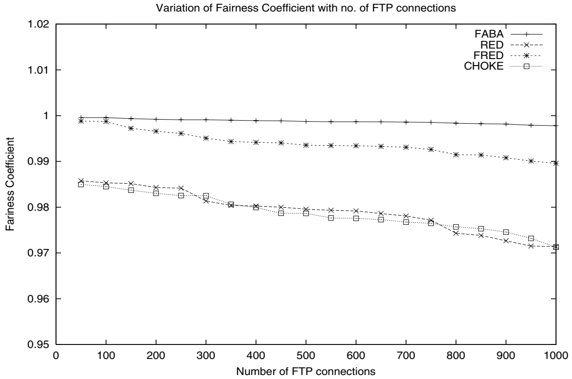


Fig. 6. Fairness coefficient versus number of FTP connections for different AQM schemes

We plot the fairness coefficient versus the number of connections and study three different applications (HTTP, TELNET, FTP), all of which use TCP as the transport layer protocol. The results in this section enable us to examine how our proposed algorithm scales with the number of connections.

In the simulation, we have a set of HTTP clients. Each client initiates several HTTP connections one by one, each connection being 5 seconds long. Hence, with a 100 second simulation time, each client performs 20 HTTP transfers. At the end, we collect the throughput obtained by each client and calculate the fairness coefficient. The simulation topology is shown in Figure 3 with the difference that all the flows are now TCP flows.

Since the Internet traffic is predominantly HTTP, it is useful to study how well a buffer management algorithm performs with HTTP traffic. Figure 5 shows how the fairness coefficient varies as the number of HTTP clients is increased. It can be seen that the fairness index is the largest (close to 1) with our proposed algorithm and is better than other AQM mechanisms. This is true even when the number of HTTP connections are large (equal to 1000).

We plot the fairness index versus the number of FTP connections in Figure 6. The fairness index versus the number of TELNET connections follows a behaviour similar to that seen in Figure 6. FABA performs consistently better than the other AQM mechanisms across a wide range of connections.

3.4 Protection for Fragile Flows

Flows that are congestion aware, but are either sensitive to packet losses or slower to adapt to more available bandwidth are termed fragile flows [2]. A TCP connection with a large round trip time (RTT) and having a limit on its maximum congestion window fits into this description.

We study FABA and other AQM algorithms with a traffic mix of fragile and non-fragile sources. The simulation scenario is shown in Figure 7. The TCP

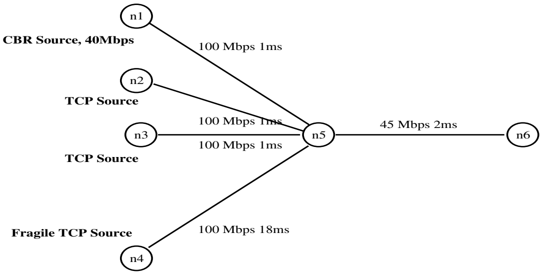


Fig. 7. Simulation scenario with a fragile flow

source originating at node $n4$ is considered a fragile flow due to its large RTT of 40 ms, while the RTTs for the other flows is 6 ms. The CBR flow sends data at a rate of 40 Mbps. Since there are 4 flows in the system and the bandwidth of the bottleneck link is 45 Mbps, ideally each flow should receive its fair share of 11.25 Mbps. We vary the maximum allowed window size, w, for the fragile flow and observe the throughput obtained by this flow. The maximum achievable throughput is then given by $\gamma_{max} = S(w/RTT)$ where S is the packet size and RTT is the round trip time. The maximum throughput is thus a linear function of the maximum window size. Ideally, the maximum throughput should never exceed 11.25 Mbps, i.e., it should increase linearly until 11.25 Mbps, and should then stabilize at 11.25 Mbps. This ideal bandwidth share is shown in Figure 8.

As can be observed from Figure 8, FABA provides bandwidth allocation for the fragile flow almost as good as in the ideal case. For a small maximum window size, every algorithm is able to accommodate the bursts of the fragile flow without any drops, but with increasing maximum window size, packet drops result in drastic reduction of the fragile flow throughput. A packet drop is fatal for a fragile source as it is slow in adapting to the state of the network. We see that the throughput becomes constant after a while since the window size of the fragile source is not able to increase beyond a threshold. Therefore, no matter how large the maximum window size is increased beyond this threshold the throughput does not increase and thus approaches a constant value. This constant value is much less than its fair share 11.25 Mbps due to the less adaptive nature of fragile flows.

4 Conclusion and Future Work

We have proposed Fair Adaptive Bandwidth Allocation (FABA), a rate control based active queue management discipline that is well suited for the network edges or Internet gateways (e.g., the ISPs). FABA achieves a fair bandwidth allocation amongst competing flows with a mix of adaptive and non-adaptive traffic. It is a congestion avoidance algorithm with low implementation overheads. FABA can be used to partition bandwidth amongst different flows in proportion to pre-assigned weights. It is well suited for bandwidth allocation among flow

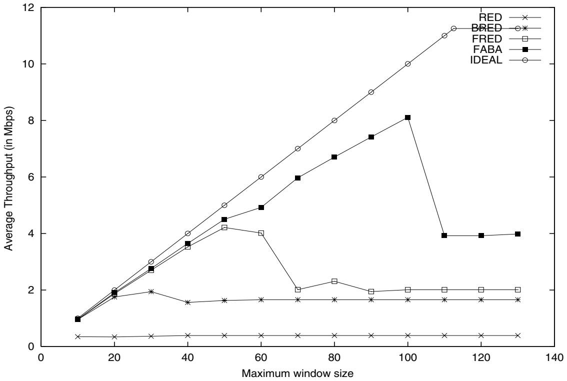


Fig. 8. Performance of the fragile flow with increasing receiver window constraint (gateway buffer size = 96 packets)

aggregates as well as for bandwidth sharing within a hierarchy as required in the differentiated services framework (see [7]). Through simulations, we have compared FABA with other well known congestion avoidance algorithms and have seen that FABA gives superior performance. FABA is shown to give high values of fairness coefficient for diverse applications (FTP, TELNET, HTTP).

A number of avenues for future work remain. It will be interesting to analytically model FABA. We need to study FABA with different topologies and with a very large number of connnections (of the order of hundreds of thousands of flows). We are currently exploring the tradeoffs between time and space complexity for the FABA algorithm.

References

1. Floyd, S., Jacobson, V.: Random Early Detection Gateways for Congestion Avoidance, IEEE/ACM Transactions on Networking, Vol. 1, No. 4, August 1993.
2. Lin, D., Morris, R.: Dynamics of Random Early Detection, In Proceedings of ACM SIGCOMM, 1997.
3. Pan, R., Prabhakar, B., Psounis, K.: CHOKe: A Stateless Active Queue Management Scheme for Approximating Fair Bandwidth Allocation, In Proceedings of IEEE INFOCOM'2000, Tel-Aviv, Israel, March 26-30, 2000.
4. Suter, B., Lakshman, T.V., Stiliadis, D., Choudhury, A.: Efficient Active Queue Management for Internet Routers, Proc . INTEROP, 1998 Engineering Conference.
5. McCanne, S., Floyd, S.: ns-Network Simulator, http://www-nrg.ee.lbl.gov/ns/
6. Kamra, A., Kapila, S., Khurana, V., Yadav, V., Saran, H., Juneja, S., Shorey, R.: SFED: A Rate Control Based Active Queue Management Discipline, IBM India Research Laboratory Research Report # 00A018, November, 2000. http://www.cse.iitd.ernet.in/∼srajeev/publications.htm
7. Kamra, A., Saran, H., Sen, S., Shorey, R.: *Full version of this paper*, http://www.cse.iitd.ernet.in/∼srajeev/publications.htm

Distributed Scheduling via Pricing in a Communication Network

Tiina Heikkinen

Department of Economics
University of Crete, Rethymno
74100 Greece
Heikkinen@econ.soc.uoc.gr

Abstract. This paper addresses the issue of pricing-based distributed resource allocation via scheduling in a communication network. Introducing the temporal aspect in the resource allocation problem presents new challenges e.g. in accounting for the durations and deadlines of the resource requests. Dynamic real-time pricing concepts are discussed for the decentralized sharing of network resources. A numerical example illustrates the benefit of congestion based pricing in a dynamic communication network. The quality of service that results from decentralized resource allocation is studied from the point of view of a power-controlled wireless network. The model is based on a dynamic noncooperative game and is related to recent work on centrally optimal (Pareto-optimal) distributed dynamic resource allocation.

Technical subject area: Scheduling, Quality of Service, Auctions, Power Management

1 Introduction

This paper addresses the issue of efficient dynamic resource allocation via distributed scheduling in a communication network. Examples of networks where resource allocation is distributed in nature include e.g. the Internet and an *ad-hoc wireless network* where the mobile users are not always directly connected to a base station. Such self-organizing networks necessitate decentralized solutions to resource allocation. Introducing the temporal aspect in the resource allocation problem presents new challenges: it becomes necessary to explicitly account for the specific deadlines and durations of the resource requests made by independent users.

The decentralized resource allocation in a communication network has been recently discussed in terms of an efficient combinatorial auction [12]. This paper presents an alternative approach to centrally optimal, i.e. *Pareto-optimal decentralized dynamic resource allocation* focusing on the allocation of a divisible network resource such as bandwidth in a communication network. Previously, [10] has illustrated how *game theory* can be used to analyze decentralized resource allocation in congested networks. This paper discusses how dynamic game theory applies in the analysis of the distributed resource allocation in a communication

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 850–862, 2002.

network. Recently, in [2] an analogous model for optimal dynamic decentralized resource allocation is presented; here like in [2] the approach is based on reducing a dynamic game to an auxiliary static game. The application to distributed bandwidth allocation in a communication network presented in this paper can be seen as a special case of the framework in [2].

This work applies a model for distributed dynamic resource allocation (cf. [2]) and focuses on the optimal quality of service QoS in a distributed dynamic network and the mechanisms through which the optimal QoS can be reached. Congestion based pricing is shown to constitute an optimal control in a decentralized dynamic network. The QoS of user i is measured in terms of the share of bandwidth allocated to user i. An example of such QoS measure is the signal-to-noise ratio in a power-controlled wireless network, see appendix A. The organization of the paper is as follows. Section 2 introduces the scheduling problem and presents the special case of a static network in 2.1. In 2.2 a general model for dynamic resource allocation is summarized; it is argued that the application to a communication network results as a limiting special case of the model in [2]. Section 3 presents models for dynamic distributed resource allocation in a communication network and discusses the connection of a pricing-based allocation model to auction games. Section 4 discusses optimal dynamic pricing and the connection between the asynchronous static network that appears as a dynamic network and the dynamic network that can depict an asynchronous "static" network (cf. [1]). The dynamic resource allocation problem in a congested communication network is related to the general load sharing problem in a capacitated network.

2 A Model of the Scheduling Problem

This paper discusses game theory under incomplete information in a setting where m users share bandwidth in a time-varying network. The general *dynamic resource allocation problem* can be defined in terms of the following elements:

- a set of T discrete time slots,
- a set $M = \{1, ..., m\}$ of m players (users) and a competitive seller (network), each with a concave payoff function $w_i, i = 1, ..., m+1$,
- resource prices $\mathbf{p} = (p_1, ..., p_T)$ at the different periods.

If the resource request for all users consists of one time slot, the scheduling problem is called single-unit (discrete) scheduling [12]; under multiunit scheduling the users may request multiple slots. In this paper the users request a part of bandwidth each time period. This part may or may not correspond to a single period or slot. Let x_{it} denote the resource allocated to user i at period t. The ith user's duration D_i is defined as: $D_i = \sum_{t=0}^{T_i} J_t(x_{it})$, where $J_t(x_{it}) = 1$ if $x_{it} > 0$ and $J(x_{it}) = 0$ otherwise. User i's resource request $(x_{it})_{t=1}^{T_i}$ is defined by its duration, its deadline $d \leq T_i$ and its value w_i:

$$w_i = \sum_{t=0}^{T_i} b^t u_{it}(x_{it}) - c_{it}. \tag{1}$$

The discount factor is denoted by $b \in (0,1)$. The ith user gets no value unless the request is satisfied before the deadline d_i. For this, $c_{it} = 0 \; \forall t < d_i$, $c_{id_i} = \sum_{t=0}^{d_i} b^t u_{it}$ and $c_{it} = b^t u_{it}$ when $t > d_i$.

Definition 2.1. *A decentralized scheduling system is defined as a system where each user $i = 1,..,m$ solves* $\max \; E[w_i]$ *such that $x_{it} \leq R_{it}$, where E denotes the expectation operator and where the resource constraint R_{it} is given.*

Let $h(\mathbf{u}_t) = \sum_{i=1}^{m}[u_{it} - c_{it}]$ denote an aggregating function; a centralized system maximizes the aggregated objective function

$$\max \; E[\sum_{t=0}^{T} b^t (\sum_{i=1}^{m} u_{it} - \frac{c_{it}}{b^t})] = \max \; E[\sum_{t=0}^{T} b^t h(\mathbf{u}_t)] \tag{2}$$

subject to $\sum_{i=1}^{m} x_{it} \leq R_t \; \forall t$, where R_t denotes the resource constraint at period t (cf. [2]). The mechanism design problem is to find prices such that the decentralized (noncooperative) solution to the scheduling problem coincides with the centralized (cooperative) solution. The issue of optimal dynamic pricing is taken up in section 4.

Let g_i denote the network link coefficient of user i [13]. Let $\alpha_i = \frac{y_i}{I_i}$ denote the quality of service QoS of user i in the network, where $y_i = g_i x_i$ denotes the received signal of users i and I_i denotes the total interference of i. The QoS α_i can be interpreted as the signal-to-noise ratio and it is the argument in u_i. Let p denote the unit price for y. The special static case where $T = 1$, $b = 1$, $c_{it} = 0$ and $r_i = u_i(\alpha) - py$ has been studied in [6] and is discussed below in 2.1 and 3.3. The connection of this simplified model to distributed dynamic resource allocation is summarized in section 4.

2.1 Distributed Resource Allocation in a Static Communication System

Recently, decentralized resource allocation in has been studied within the framework of general equilibrium theory in [8] and [12] ("the market-oriented programming approach") for the Internet and in [6] for a wireless communication network. In a static (one period) wireless network a number of users m need to share the same time slot. Define the quality of service (QoS) in terms of signal-to-noise-ratio (SNR) in a power controlled wireless network [13]. Letting the users be the players and the signal-to-noise ratio of a user define the payoff of the user, the SNR-model defines an externality game [5].

Definition 2.2. *An externality game (EG) is a noncooperative game defined by m players with submodular payoff functions $u_i, i = 1,..,m$.*

The key restriction on a representative user i's payoff SNR $u_i = \alpha_i$ is that it depends on his own strategy, here in terms of received signal power y_i, and the externality vector $(y_{j \neq i})$ through $I = I(\mathbf{y})$. Let g_i denote the link fading coefficient for user i; thus, define (cf. [13,6])

$$u_i = \alpha_i(x_i, I(\mathbf{y})) = \frac{W g_i x_i}{I(\mathbf{y})} = \frac{g_i x_i}{\sum_{j \neq i} g_j x_j} = \frac{y_i}{\sum_{j \neq i} y_j} \tag{3}$$

where $W = 1$ denotes the given bandwidth. Here α_i is a *submodular function* i.e. function with decreasing differences: for all $\mathbf{y}, \mathbf{y}'$ in the strategy space and $\mathbf{y} > \mathbf{y}'$,

$$\alpha_i(\mathbf{y}, I) - \alpha_i(\mathbf{y}', I) = \frac{y_i - y_i'}{\sum_{j \neq i} y_j} \tag{4}$$

is nonincreasing in I. Thus, a communication network with limited bandwidth is an example of an externality game motivating externality, i.e. congestion pricing. A non-atomic game is a game with a continuum (large number) of users, in which no single player can affect the other players by changing only her own action. The impact of a single user on the total interference is negligible if the number of users is large. The users as noncooperative players ignore the impact of their usage of resource (transmit power) on the congestion externality, interference, caused to other users.

Recently [6] has discussed two approximately equivalent max-min fair congestion based pricing strategies in an interference limited wireless system:

- *QoS* pricing consisting of defining a congestion based unit price $p_\alpha = pI,\ p > 0$ for *QoS* α (based on [4]);
- Pricing for received signal power y consisting of a strictly positive unit price p for y.

The equivalence is based on considering the first order optimization condition with respect to y_i under power-based pricing, given I_i (cf. current $CDMA$):

$$\frac{du_i}{dy_i} = \frac{1}{I_i} u'(\frac{y}{I_i}) - p = 0, \tag{5}$$

equivalent to the first order condition under *QoS*-pricing:

$$u'(\alpha_i) = pI = p_\alpha. \tag{6}$$

Here I denoted interference caused, and when $I \approx I_i$ (interference received), the equivalence holds. Based on this, consider only power-based pricing in what follows.

2.2 A General Model for Distributed Dynamic Resource Allocation

Sometimes it is assumed for simplicity that $I(\mathbf{y}) = \sum_{i=1}^m g_i x_i = \sum_{i=1}^m y_i$ (cf. [8]). I.e. the utility function of a representative agent i is given by

$$u_{it} = \frac{y_{it}}{\sum_{i=1}^m y_{it}}. \tag{7}$$

The form in (7) is a special case of the parameterized family of utility functions in [2] (where $c_{it} = 0\ \forall i, t$):

$$u_{it} = \frac{y_{it}}{h(\mathbf{y})^{1-\beta}} \tag{8}$$

where $\beta \in (0, 1)$ and h denotes the aggregating function, assumed to be convex and homogeneous of degree 1. The aggregating function $h(\mathbf{u})$ for a given time period then satisfies the following [2] equality:

$$h(\mathbf{u}(\mathbf{y})) = h(\mathbf{y})^{\beta}. \tag{9}$$

Note that the utility functions in (8) define an externality game.

Proposition 2.3. *The utility function (7) for a user in a congested communication network (e.g. interference-limited wireless network or Internet) corresponds to the case of the family of functions (8) when $\beta = 0$. The aggregating function h satisfies (9) also in this case.*

Proof. Letting $\beta = 0$ and $h(\mathbf{y}) = \sum_{i=1}^{m} y_{it} = I(\mathbf{y})$ in (8) yields (7). Also, (9) is satisfied:

$$h(\mathbf{u}(\mathbf{y})) = \sum_{i=1}^{m} \frac{y_{it}}{\sum_{i=1}^{m} y_{it}} = 1 = (\sum_{i=1}^{m} y_{it})^0 = h(\mathbf{y})^{\beta} \; \forall t. \tag{10}$$

The limiting case when $\beta = 0$ is not addressed in [2]. Simplified solutions to the distributed dynamic resource allocation problem corresponding to centrally optimal Nash equilibria are discussed next.

3 Solution Concepts in a Dynamic Network

3.1 Social Nash Equilibria in the Non-cooperative Network Game

Consider the model in 2.2 for distributed resource allocation. Assume that the users have utility functions w_i where u_{it}s are as in (7). Let $d = T_i = T$ be the deadline for all agents. Each user i has initial endowment R_{i0}. At each stage the user should divide his resource into

- x_{it} for obtaining immediate gain α_{it} given g_{it} and the choices of other users
- $z_{it} = R_{it} - x_{it}$ for savings. Let $R_{it+1} = z_{it}$ (the "cake-eating" problem in [2]).

The model of resource allocation may be regarded as a non-cooperative dynamic stochastic game [2]. The main solution concept in noncooperative network games (cf. [8]) is a Nash equilibrium [9].

Proposition 3.1. *The centrally optimal Nash equilibrium strategy is to use the share of resources depending only on the remaining time periods:*

$$x_{it}^{*} = R_{it}(\frac{1-b}{1-b^{T-t+1}}). \tag{11}$$

Proof. Analogous to the proof in [2] for $\beta \in (0, 1)$.

I.e. the results in [2] hold also when $\beta = 0$ in (8), the case corresponding to the utility function model in a congested communication network ([8,6]). Equation (11) indicates that the user becomes more impatient as the deadline T approaches.

Section 3.2 presents a simplified example for a social Nash equilibrium, an application of the general model above to a distributed network where the users are allowed to decide each period only whether to participate or to opt out. I.e. $x_{it} \in \{0,1\}$, for all the iterations t for all users i, and the comparison is between $\alpha_{it} = \frac{g_{it}}{\sum_{j=1}^{m} g_{jt}}$ and $bE(\alpha_{it+1}) = bE(\frac{g_{i,t+1}}{\sum_{j=1}^{m} g_{j,t+1}})\ \forall i, \forall t$. The relation between the dynamic model in 3.2 and an analogous model in 3.3 is summarized in section 4 from the point of view of optimal dynamic pricing.

3.2 A Delay-Limited Solution to Resource Allocation

The birth-death process of new users entering and users whose deadlines passed leaving make the channel time-varying. Alternatively or simultaneously the link coefficients (g_is) are random variables. In a time-varying channel with a time cost, the participation externality game can by described as follows.

A Centralized Solution to Dynamic Resource Allocation

Let $y_i = g_i x_i$ denote the received signal, where $g_i = (\mathbf{G})_{ii}$ is the fixed link coefficient and x_i denotes the power allocation of i. Let $f_{ij} = (\mathbf{F})_{ij} = g_j$ be the coefficient of interference received by user i from j [13]. Let $\mathbf{y} = (y_{i0}, .., y_{iT})_{i=1}^{m}$ and let the duration be equal across users $T_i = T\ \forall i$. The dynamic resource allocation problem can be stated as, when taking into account the delay costs formalized by $b \in (0,1)$ and an additive deadline cost c_t, equal across users (cf. [5]),

$$\max_{\mathbf{y}} E \sum_{t=0}^{T} b^t \alpha_t \quad s.t. \quad \mathbf{F}_t \mathbf{x}_t \alpha_t = \mathbf{G}_t \mathbf{x}_t. \tag{12}$$

The QoS of i is measured by $\alpha_i = \frac{(\mathbf{Gx})_i}{(\mathbf{Fx})_i}$. If $g_i = g = 1$ is fixed, $\alpha_t = \frac{1}{m_t}\ \forall t$ (Appendix A); only the number of users in the network varies over time. Therefore, letting $x_{it} = y_{it} \in \{0,1\}\ \forall i,\ \forall t$, the scheduling problem becomes a *load-balancing* problem over time when the network coefficients are fixed. In general, the network coefficients are random variables. Let $\mathbf{e}$ denote the m-vector of ones. In what follows assume the randomness in α is also due to time-varying fading conditions as captured by the random link coefficient g, initially letting $\mathbf{x} = \mathbf{e}$: $\alpha_i = \frac{g_i}{\sum_{i=1}^{m} g_i}\ \forall i$ where also m can be a random variable.

Let there be two state variables, α and s: $s = 0$ if α for the representative user (omitting user index) is currently not in the system and $s = 1$ if the user participates in the network. If the randomly chosen user accepts current α_t he stays in the system for period t, is absent period $t+1$ and makes a new decision at $t+2$. The Bellman equations for $v(\alpha, s), s = 0, 1$ are, assuming there is in addition to the deadline delay cost defined by c a proportional delay cost component $b \in (0,1)$ when deferring the choice of the resource usage to the next period $t+1$

$$\begin{aligned} v(\alpha, 0) &= \max\{\alpha + bEv(\alpha, 1), bEv(y, 0) - c\} \\ v(\alpha, 1) &= \max\{bEv(y, 0)\}. \end{aligned}$$

Define the threshold $\bar{\alpha}$ as in optimal stopping problems by

$$\bar{\alpha}_t + bEv(\bar{\alpha}_t, 1) = bEv(y, 0) - c_t, \tag{13}$$

where $\bar{\alpha}$ is the threshold QoS. Note that $\frac{\sum_{i=1}^{m} Ev(\alpha_i, s)}{m} = Ev(\alpha, s)$ assuming the g_is are independent and identically distributed random variables. Let this model define the expected value of QoS as [5]:

$$Ev(\alpha, 0) = \frac{\bar{\alpha}_t(1+b)}{b(1-b)} + \frac{c_t}{b(1-b)}. \tag{14}$$

The optimal threshold $\bar{\alpha}$ can be characterized as: $\bar{\alpha}_t = \frac{E(\alpha) - \int_0^{\bar{\alpha}} \alpha f(\alpha) d\alpha - \frac{c_t}{b}}{\frac{1}{b} + (1 - F(\bar{\alpha}))}$ where F is the cumulative density function of α and f denotes the distribution function of $\alpha = (g_i)/(\sum_i g_i)$.

Noncooperative Equilibrium in Dynamic Resource Sharing

It can be shown that the centralized model summarized by equations (3.2) is equivalent to a simplified version of the noncooperative model in [2] with Pareto-optimal outcomes, where the corresponding utility function of user i in a one-shot auxiliary game for strategy profile $\mathbf{x} \in \{0,1\}^m$ is:

$$r_i = \frac{x_i}{I(\mathbf{x})} + b\frac{(1 - x_i)}{I(\mathbf{e} - \mathbf{x})}, \tag{15}$$

where $R_{it} = 1 = e_i \; \forall i$ (cf. 3.1). The first order condition (f.o.c) of user i is to let $I((\mathbf{x})^*) + x_i^* \frac{dI(\mathbf{x})^*}{dx_i} = b(I(\mathbf{e} - \mathbf{x}^*) + (e_i - x_i^*)\frac{dI((\mathbf{e} - \mathbf{x}^*)}{dx_i})$. By the homogeneity of I, $\sum_{i=1}^{m} x_i \frac{dI(\mathbf{x})}{dx_i} = (\beta - 1)I(\mathbf{x}) = -I(\mathbf{x})$. When summing the f.o.c.s this implies that at the deterministic Nash equilibrium $I(\mathbf{x}_t^*) = bI(\mathbf{x}_{t+1}^*)$ equivalent to: $\alpha_t = b\alpha_{t+1}$, as in the load-balancing Pareto-optimal solution. The corresponding $\bar{\alpha}$ is defined by m_t such that: $\bar{\alpha} = \frac{1}{m_t} = \frac{b}{m - m_t}$.

The above centralized formulation also directly captures a noncooperative model, due to the symmetry across users in terms of the parameters b, c and u_i (identity). The noncooperative users simultaneously or asynchronously pick a strategy x_{it} from $A = \{0,1\}$ (or, if only m varies, $y_{it} \in A$) each period t when the QoS was rejected at $t-1$ or accepted at $t-2$. The centralized case above assumed that the distribution f of α is stationary. This need not be the case in a time-varying network. However, assume that the environment becomes stationary after some finite time. Then, as argued in [5], a simple (participation) EG such as defined above (by $card(A) = 2$) is D-solvable which implies that users in the linear utility model will converge to a unique Nash-equilibrium even in noisy distributed asynchronous settings. The argument is as follows. Given the ultimately stationary distribution of the number of users, the strategy choices of the other users in terms of α and $Ev(\alpha)$ appear as parameters in the participation decision.

Note that formally the maximization of α is equivalent to the maximization of $b\alpha - py$, letting utility parameter b satisfy: $b = 1 + p_\alpha$, where $p_\alpha = Ip$. By tuning the price the QoS in the network can be maintained at the desired level.

3.3 The “Static” Network Can Be Seen as a Dynamic Game

The iterative solution of the static distributed resource allocation has connections to a dynamic game.

Definition 3.2. *A game is said to be dynamic if at least one of the players has more than one information set [1].*

An *information set* is a concept for games in extensive (tree) form [1]: the same strategy set is available at each node in the same information set, where the nodes correspond to possible events; no node follows another node in the same information set.

Consider the following iterative solution of a resource (bandwidth=1) sharing game in a wireless system where $T = 1$ [6].

Definition 3.3. *Let $A_i = \{0, 1\}$ denote the strategy set [1] of user $i = 1, .., m$. An asynchronous greedy algorithm for the game defined by the payoff functions*

$$r_{it} = u_{it}(\alpha_{it}) - py_{it},\ i = 1, .., m, \tag{16}$$

where $\alpha_{it} = \frac{y_{it}}{I_{it}+e}$, $I_{it} = \sum_{j\,/\,\neq i} y_{jt}$, $e = \alpha n = 0.0001$ (see Appendix A) and the u_{it}s are given in (7), consists of iteratively finding, for $i = 1, ..., m$,

$$y^*_{it} = \arg\max_{y_{it} \in A_i} u_i\left(\frac{y_{it}}{I_{it} + e}\right) - py_{it}, \tag{17}$$

The above greedy algorithm defines a solution to *bandwidth-scheduling* under a price penalty when $R_{it} = 1\ \forall i$ (i.e. the resource constraints are due to $A_i = \{0, 1\}\ \forall i\ \forall t$). The users opting in at period t set $y^t_i = 1$. The corresponding QoS price at period t is $p_\alpha = pm_t$, where m_t denotes the number of users present in the network at period t.

Consider the tree representation of the greedy algorithm. At period $t = 1$, a user i is picked randomly to update its resource demand given I_1 (given the initial values of the other users choices). The corresponding node in the game tree is the root of the tree. The following period $t = 2$ another user from the set M is chosen randomly to update its demand, given I_2. Iterating the tree construction yields a *binary tree* where each user faces the information set consisting of all nodes (events): given the total interference I_t, the user is unable to determine the decomposition of I_t, i.e. which other users are present.

The sequential solution of the originally static game thus yields a dynamic game where each user i will make a decision (reach an information set) k times where k is the time per user that the greedy algorithm takes before converging to a Nash equilibrium. By the concavity of the payoff functions such an equilibrium exists. The reason for the ”static” game to appear as a dynamic game is that each player acts more than once and hence time plays a role (cf. [1]). Figure 1 depicts the total time required for the greedy algorithm in the special case of *linear utility model* where $u_i(\alpha_i) = b_i\alpha_i = i\alpha_i\ \forall i$, $m = 20$, $p = 1$. Even when adding the dominated strategies ([9]) $\{0.5, ..., \frac{1}{20}\}$ the greedy algorithm converges to the Nash equilibrium in less than five iterations per user. Only the users with high utility participate in the network, i.e. users $i = 11, ..., 20$, yielding the sum of utilities 17.22 at the Nash equilibrium (which can be easily verified).

In the absence of a price $p = 0$ the sum of utilities is lower (11.05). At $p = 19$, the sum of utilities is at highest 39. But for users $i = 9, ..., 18$ this equilibrium yield less than the equilibrium under $p = 1$ and hence does not constitute a Pareto-improvement from the Nash equilibrium at $p = 1$.

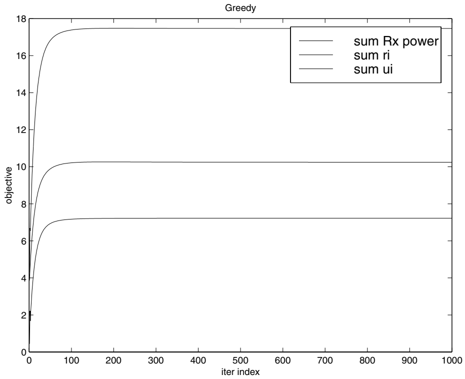


Fig. 1. Greedy Algorithm: Sum of Utilities (topmost curve), Received Powers (middle curve) and Net Utilities (lowest curve) for $m = 20$, $p = 1$, $u_i = i\alpha_i$, $A_i = \{1, 0.5, ..., \frac{1}{20}, 0\}$, $i = 1, .., m$, $e = 0.0001$.

Dominance Solvable Network Games

In the linear utility example in section 2 where $u_i(\alpha_i) = i\alpha_i$, the middle strategies are strictly dominated [9]. In a linear model it suffices to consider participation games where $A_i = \{0, 1\}$, $i = 1, .., m$. Participation games constitute an important class of externality games EGs. The main result for participation EGs is related to their dominance-D-solvability [5].

Definition 3.4. *Let $u_i = \alpha_i$. For a given player i, strategy $\mathbf{y}$ dominates strategy $\mathbf{y}'$ relative to strategy set A iff $\alpha_i(\mathbf{y}, I) > \alpha_i(\mathbf{y}', I)$ for given I. The set of strategies which are not dominated are denoted by $D_i(A)$.*

Define $D^k(A) = D(D^{k-1}(A))$ with $D^1 = D(A)$. Note that $D^\infty = \lim_{k\to\infty} D^k$ is well defined due to the monotonicity of D.

Definition 3.5. *A game is D-solvable if $card(D^\infty) = 1$ for all i.*

D-solvability (solvability by dominated strategies) requires that given only partial information of the opponent's strategies as captured by I_{it}, the player still can eliminate dominated strategies.

Proposition 3.6. *Let $A_i = \{0, \frac{1}{2}, .., \frac{1}{m}, 1\}$ $\forall i$. The received power allocation game defined by m payoff functions $r_i = \alpha_i - py_i$, where $\alpha_i = \frac{b_i y_i}{I_i + e}$ where $b_i = i$ is $D - solvable$.*

Proof. Given I_{it}, e and p, either $\frac{b_i}{I_{it}+e} - p > 0$ implying $y^*_{it} = 1$ or $\frac{b_i}{I_{it}+e} - p \leq 0$ implying $y^*_{it} = 0$. I.e. for any t and I_{it} one of the $y^*_{it} = 1$ or $y^*_{it} = 0$ strategies is better than the other strategies and $D^\infty \subseteq \{0,1\}$ $\forall i$. For all $i = 1,..,i_{th}$ $y^*_i = 1$ and $y^*_i = 0$ for $i < i_{th}$ where the threshold user i_{th} is defined as the upper integer part of the solution i_{th} to $\frac{i_{th}}{m-i_{th}-1+e} = p$:

$$i_{th} = \lceil \frac{p(m-1+e)}{1+p} \rceil. \tag{18}$$

In the example the greedy algorithm converges to a unique Nash equilibrium. In general, an equilibrium exists in a noncooperative game assuming that the utility functions are concave [9].

Proposition 3.7. *A Nash equilibrium in the resource allocation game defined by the utility functions (16) is Pareto-optimal.*

Proof. Consider a given Nash equilibrium allocation $\mathbf{y}_t$. The utility of the ith participating agent is $r_{it} = \frac{b_i y_{it}}{\sum_{j \neq i} y_{jt} + e} - p y_{it} \geq 0$. Given b_i and p, r_{it} can not be increased by increasing y_{it} without decreasing $r_{jt} = \frac{b_j y_{jt}}{\sum_{k \neq i,j} y_{kt} + e} - p y_{jt}$ $\forall j \neq i$.

A Relation to Auctions

An alternative solution concept for a dynamic resource sharing game is to consider specific auction protocols. Recently, in [12] an ascending auction protocol for the general discrete resource allocation problem is presented. There is a connection between the auction approach and the linear utility model under a given price p. By specifying his utility function u_i user i makes an implicit bid consisting of the amount that he is willing to pay at maximum for one unit of transmission. When $I = \sum_{i=1}^{m} y_i$, $b_i = i$ and $y_i \in \{0,1\}$ $\forall i$, the optimal price inducing at most one user to be present at any time period t under the greedy algorithm is $p^*_t = \max_i \; i \in \{1,..,m_t\} = m_t$.

4 Optimal Dynamic Pricing and the Connection between a "Static" Network and a "Dynamic" Network

A dynamic network as described in section 3.2 above explicitly defines the resource usage problem for a representative user for T time periods. The asynchronous resource sharing game in section 3.3 is concerned with the iterative solution for a "static" resource sharing problem. However, there is a direct relation between the two models.

Under "fictitious play" [3], the users behave as if they think they are facing a stationary (but unknown) distribution of opponent's strategies. The model in section 3.2, where in general the number of users m is a random variable, captures this idea: each period the representative user behaves so as to maximize the expected payoff. In what follows let $0 \leq \beta = \frac{b(1-b)}{1+b} < 1$ and for now assume that the additive delay cost is zero by letting $c = 0$. The user participates in the network at period t letting $y_t = 1$ if

$$\alpha_t \geq \frac{b(1-b)}{1+b} Ev(\alpha, 0) = \beta Ev(\alpha, 0) = \bar{\alpha}. \tag{19}$$

In the "static" linear utility model in section 3.3 the utility of user i at period t was defined as $r_{it} = i\alpha_{it} - py_t = u_{it}(\alpha_{it}) - py_{it}$. Here consider a symmetric version of this linear model and let $u_{it}(\alpha_{it}) = b'(\alpha_{it})$. In the symmetric linear model the representative user participates in the network at period t letting $y_t = 1$ if

$$\alpha_t \geq \frac{1}{b'} p. \tag{20}$$

The decision rules in (19) and (20) coincide if

$$\beta = \frac{1}{b'} \tag{21}$$

and the price reflects the expected value of QoS

$$p = Ev(\alpha, 0). \tag{22}$$

In a symmetric dynamic Nash equilibrium the users can find the optimal strategy fast applying the greedy algorithm under optimal pricing: $p = Ev(\alpha)$.

The relation (21) can be interpreted as follows. The representative user in the linear utility model is more demanding if b' is relatively high; this is analogous to a low value of β in a temporal allocation model: the greedier user is also the more impatient.

Above letting $c_t = 0 \; \forall t$ reflects infinite deadline for simplicity. In the presence of a deadline cost, i.e. when $c_{id} > 0$,

$$\bar{\alpha}_{d_i - 1} = \max\{Ev(\alpha, 0)\frac{b(1-b)}{1+b} - \frac{c_{id}}{1+b}, 0\} < \bar{\alpha}. \tag{23}$$

I.e. when the deadline period d_i is ahead, the acceptance threshold will be lower. Analogously to (22)and (23), the price before deadline becomes

$$p_{d_i - 1} = \max\{Ev(\alpha, 0) - \frac{c_{id}}{b(1-b)}, 0\}. \tag{24}$$

Assuming $p = Ev(\alpha, 0) - \frac{c_d}{b(1-b)}$ as in (24) at period before deadline $d - 1$, and $p = Ev(\alpha, 0)$ otherwise, relates the optimal (QoS-price equivalent) price p to the expected value of QoS (at state 0) in a symmetric network. For the user to participate, the price in terms of opportunity cost must not be greater than the benefit from participation. In an asymmetric network the different users may have different deadlines corresponding to different $\bar{\alpha}_i$s and the users $i = 1, ..., m$ iterate the following Bellman equations:

$$\max_{y_{it} \in \{0,1\}} \{0, b_i \alpha_{it}(y_{it}) - \bar{\alpha}_{it}\}, \; t = 0, .., T_i. \tag{25}$$

The corresponding greedy algorithm successively eliminates dominated strategies in the dominance solvable game (cf. Proposition (3.6)) for users $i = 1, ..., m$

$$\max_{y_{it} \in (0,1)} b_i \alpha_{it}(y_{it}) - \bar{\alpha}_{it} y_{it} \tag{26}$$

where $\alpha_{it} = \frac{y_{it}}{I_{it}} = \frac{y_{it}}{m_{it}}$. The price p_{it} is fixed at $p = Ev(\alpha)$ until the deadline period $t = d_i - 1$ of i is reached at which point the price becomes zero if $p_{d_i - 1} = 0$ in (24). Consider the numerical example for the greedy algorithm in section 3.3 where the Nash equilibrium is found in less than five iterations per user such that users $i = 1, .., 9$ converge to setting $y_i^* = 0$. However, suppose the deadline d_i for these users is after period 5, e.g. period 6 whereas the deadline for the users $i = 10, .., 20$ with duration five is at period 5 after which they will disappear from the system. Then, the optimal dynamic equilibrium allocation is given by $y_{it} = 1, i = 10, .., 20$, $t = 1, .., 5$ and $y_{it} = 1$ for $i = 1, .., 9$ at $t = 6$ and zero otherwise. Since the users update their strategies in a random order, this equilibrium is an approximately fair efficient solution to dynamic resource allocation.

The *socially optimal Nash equilibrium in the network externality game* as defined by utility functions in (8) can be rationalized by an implicit congestion based price for QoS, equivalent to a unit price p on y a argued in 2.1. A similar observation is pointed out in [2]: if we regard the impact of the externality h on the individual utility as a resource price, then the individual utilities should be proportional to $h^{\beta-1}$. The network externalities are internalized when the payoff functions are as in (8). Likewise, if the QoS requests are feasible, optimal power control can be distributed [6].

5 Conclusion

The focus in this paper has been on centrally optimal distributed scheduling in a communication network with a divisible resource, bandwidth. In some applications the acceptable QoS level implies that only a single user can be allocated to a given time slot; in the framework of the current paper there is a price level (cf. recent related work [12]) that corresponds to allowing only one user at a time to be present.

An asynchronous resource sharing game over time defines a dynamic game. The scheduling problem adds user-specific durations and deadlines. The uncertainty under dynamic resource allocation comes from the time-varying load (number of users) in the network. An alternative interpretation is to consider time-varying link coefficients.

Appendix A

The centralized uplink power control problem [13] in a single-cell network can be summarized as follows. Let α denote the QoS requirement in terms of the signal-to-noise ratio. Let x_i denote transmit power allocation of user i, let g_{ij} denote the fading coefficient between mobile i and base j and let n be the additive Gaussian noise. The power control problem [13] is to find $\mathbf{x}$ such that $\mathbf{Gx} = \alpha(\mathbf{Fx} + \mathbf{n})$, $\sum_i x_i \leq R$, where $\mathbf{F}$ denotes the interference (externality) matrix and R denotes the resource constraint. When $n = 0$, a solution is to let $x_i = \frac{1}{g_i}$ and $\alpha_i = \frac{g_i x_i}{\sum_{j \neq i} g_j x_j} = \alpha = \frac{1}{m-1}$ $\forall i$, where m denotes the number of users.

References

1. T. Basar and G. Olsder: Dynamic Noncooperative Game Theory. Academic Press (1995)
2. V. Domansky and V. Kreps: Social Equilibria for Competitive Resource Allocation Models. proc. Constructing and Applying Objective Functions, Springer Lecture Notes in Economics and Mathematical Systems (2002)
3. D. Fudenberg and D. Levine: The Theory of Learning in Games. The MIT Press (1999)
4. T. Heikkinen: Dynamic Pricing of a Multimedia Network. proc. CISS, Princeton (1998)
5. T. Heikkinen: On Learning and Quality-of-Service in a Wireless Network. proc. Networking, Paris, France (2000)
6. T. Heikkinen: On Congestion Pricing in a Wireless Network. Wireless Networks, forthcoming
7. H. Ji and H. Cing-Yao: Noncooperative Uplink Power Control in Cellular Radio Systems. Wireless Networks 4 (1998)
8. F. Kelly and R. Gibbens: Resource Pricing and the Evolution of Congestion Control. http://www.statslab.cam.ac.uk/ frank/evol.html
9. R. Myerson: *Game Theory*. Harvard University Press (1991)
10. S. Shenker: Making Greed Work in Networks. IEEE Tr. on Networking 3(6) (1995)
11. N. Shroff, M. Xiao and E. Chong: Utility based power control in cellular radio systems. Proc. Infocom, Anchorage (2001)
12. M. Wellman, W. Walsh, P. Wurman and J. Mac-Kie-Mason: Auction Protocols for Decentralized Scheduling. Games and Economic Behaviour, forthcoming
13. Zander, J.: Performance of Optimum Transmitter Power Control in Cellular Radio Systems. IEEE Transactions on Vehicular Technology (1992) vol. 41, no. 1, Feb.

A Simulation Study of Access Protocols for Optical Burst-Switched Ring Networks

Lisong Xu, Harry G. Perros, and George N. Rouskas

Department of Computer Science, North Carolina State University, Raleigh, NC, USA
lxu2,hp,rouskas@csc.ncsu.edu

Abstract. In this paper, we consider a WDM metro ring architecture with optical burst switching. Several access protocols are proposed and their performance is analyzed by simulation.

1 Introduction

Optical burst switching (OBS) [1,2,3,4,5,6] is a switching technique that occupies the middle of the spectrum between the well-known circuit switching and packet switching paradigms, borrowing ideas from both to deliver a completely new functionality. The unit of transmission is a *burst*, which may consist of several packets. The transmission of each burst is preceded by the transmission of a control packet, which usually takes place on a separate signaling channel. Unlike circuit switching, a source node does not wait for confirmation that a path with available resources has been set up; instead, it starts transmitting the data burst soon after the transmission of the control packet. We will refer to the interval of time between the transmission by the source node of the first bit of the control packet and the transmission of the first bit of the data burst as the *offset*. The control packet carries information about the burst, including the offset value, the length of the burst, its priority, etc. Based on this information, intermediate nodes configure their switch fabric to switch the burst to the appropriate output port. However, in case of congestion or output port conflicts, an intermediate node may drop a burst. Also, consecutive bursts between a given source-destination pair may be routed independently of each other.

There are several variants of burst switching, mainly differing on the length of the offset. The most well-known scheme is *Just Enough Time* (JET) [3], in which the offset is selected in a manner that takes into account the processing delays of the control packet at the intermediate switches. Let $T_i^{(p)}$ denote the processing delay of a control packet at an intermediate switch, $T_d^{(p)}$ denote the processing delay of a control packet at the destination switch, and $T_d^{(s)}$ denote the time to setup (configure) the destination switch. Then, the offset value for JET is :

$$\text{offset}_{\text{JET}} \quad = \quad \left(\sum_i T_i^{(p)} \right) \; + \; T_d^{(p)} \; + \; T_d^{(s)} \tag{1}$$

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 863–874, 2002.

One issue that arises in computing the offset under JET is determining the number of intermediate switching nodes (hops) between the source and destination. Information about the number of hops in a path may not, in general, be readily available; even if it is known, it may not be valid when used. Thus, it is desirable to use an offset value that does not depend on the path used and does not require the exchange of information among network nodes.

The part of the offset value that depends on the path between the source and destination is the sum of the processing times at intermediate nodes. Given recent advances in hardware implementation of communication protocols, we can assume that the processing time $T_i^{(p)}$ in (1) will be very short for most common functions of the signaling protocol. In this case, fiber delay lines may be used at intermediate nodes to delay each incoming burst by an amount of time equal to $T_i^{(p)}$. Then, the first term in the right hand side of (1) can be omitted when computing the offset. We call this new scheme the *Only Destination Delay (ODD)* protocol, and its offset is given by:

$$\text{offset}_{\text{ODD}} = T_d^{(p)} + T_d^{(s)} \tag{2}$$

Instead of using destination-specific values for the processing and switching delays in (2), one may use a constant offset value by taking the maximum of these values over all destinations. Such a value may significantly simplify the design and implementation of signaling protocols and optical switches for burst switching networks [2].

In this paper we study burst switching protocols for ring networks. Our focus on ring topologies is motivated by the wide deployment of optical rings. These networks represent a significant investment on the part of carriers, and are currently being upgraded to support WDM. Section 2 describes the ring network we consider and the basic operation of burst switching in such an environment. Section 3 provides a detailed description of the various burst switching access protocols studied in this paper. Section 4 presents the simulation results on the performance of these burst switching access protocols, and finally Section 5 provides some concluding remarks.

2 The Ring Network under Study

We consider N OBS nodes organized in a unidirectional ring, as shown in Figure 1. Each fiber link supports $N + 1$ wavelengths. Of these, N wavelengths are used to transmit bursts, and the $(N + 1)$-th wavelength is used as the control channel. Each OBS node is attached to one or more access networks. In the direction from the access networks to the ring, the OBS node acts as a concentrator. Buffered packets are grouped together and transmitted in a burst to the destination OBS node. A burst can be of any size between a minimum and maximum value. Bursts travel along the ring without undergoing any electro-optic conversion at intermediate nodes. In the other direction, from the ring to the access networks, an OBS node terminates optical bursts, electronically processes the data packets contained therein, and delivers them to users.

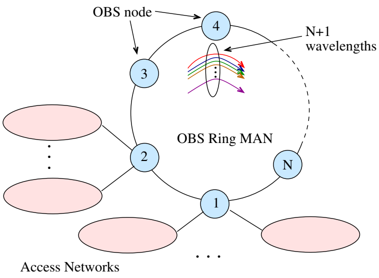


Fig. 1. OBS Ring MAN

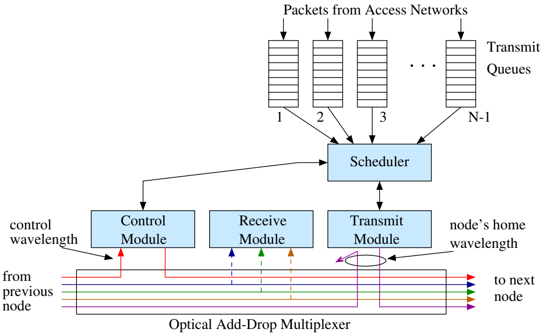


Fig. 2. Node architecture (delay lines not shown)

The architecture of an OBS node is shown in Figure 2. Each node is equipped with one optical add-drop multiplexer (OADM), and two pairs of optical transceivers. The first pair consists of a receiver and transmitter tuned to the control wavelength, and is part of the control module in Figure 2. The control wavelength is dropped by the OADM at each node, and added back after the control module has read the control information and has inserted new information.

The second pair of transceivers consists of a transmitter that is fixed tuned to the node's *home wavelength*, and an agile receiver that can receive from all N data wavelengths. Each OBS node has a dedicated home wavelength on which it transmits its bursts. The OADM at each node removes the optical signal from the node's home wavelength by dropping the corresponding wavelength, as Figure 2 illustrates. The OADM also drops the optical signal on other burst wavelengths, whenever they contain bursts for this node. In the case where multiple bursts arrive, each on a different wavelength, at an OBS node, the receive module in Figure 2 employs a collision resolution strategy to determine which burst will be accepted. To support ODD, an extra fiber delay line (not shown in Figure 2) is added into the node to delay outgoing data on all wavelengths except the control wavelength and the node's home wavelength.

Packets waiting for transmission are organized into transmit queues according to their destination. The order in which transmit queues are served is determined by the scheduler module in Figure 2. We assume that the transmit queues are considered in a Round-Robin manner.

The control wavelength is used for the transmission of control slots. In a ring with N nodes, N control slots, one for each node, are grouped together in a *control frame* which continuously circulates around the ring. Depending on the length of the circumference of the ring, there may be several control frames circulating simultaneously. In this case, control frames are transmitted back-to-back on the control wavelength. Each node is the owner of one control slot in each control frame. Each control slot contains several fields, as Figure 3 illustrates. The format and type of the fields depend on the OBS protocol used. In general, however, each control slot includes fields for the destination address, the offset, and the burst size. Other fields may be included for some of the protocols.

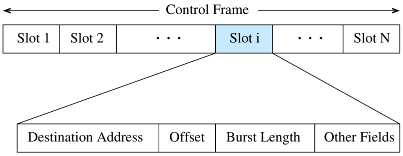


Fig. 3. Structure of a control frame

To transmit a burst, a node waits for the next control frame and writes the burst information (destination address, burst length, and offset) in its own control slot. If it has nothing to transmit, it just clears all the fields in its control slot. Each node also reads the entire control frame to determine whether any control slots indicate a burst transmission to this node. If so, and assuming that the node is not in the process of receiving another burst, it instructs its tunable receiver to tune to the appropriate wavelength to receive the burst; that is, preemption is not allowed. In case of a receiver collision (i.e., when the address of this node is specified in multiple control slots), the destination node selects one of the bursts to receive.

Each node acts as a source node (inserting bursts), as an intermediate node (passing through bursts to downstream nodes), and as a destination node (terminating bursts). As a result, each node must read each control frame in its entirety before determining what action to take. Therefore, in a ring network the time to process a control frame is the same for intermediate and destination nodes (i.e., $T_i^{(p)} = T_d^{(p)}$). The control frame is delayed by this amount of time as it passes through each node. This delay is the sum of the control frame transmission time plus the time to process the control frame, and it can be kept short by employing a simple protocol implemented in hardware. A number of OBS protocols having these features are described in the next section.

3 OBS Protocols

Since each OBS node is assigned a unique home wavelength, bursts may be lost due to receiver collisions. This occurs when two or more nodes transmit bursts to the same destination, and the burst transmissions overlap in time. We propose several access protocols which can be classified in three classes, depending on how receiver collisions are resolved.

1. *Source node.* The source node resolves receiver collisions using the information transmitted on the control wavelength.
2. *Destination node.* A source node must get permission from the destination to send a burst. Each destination schedules requests to avoid collisions.
3. *Token passing.* Tokens are used to resolve receiver collisions.

Our emphasis is on protocols that use few rules, are simple to implement in hardware and are distributed in nature. We have deliberately avoided protocols that are centralized in nature, or they require the collection of transmit queue sizes, or they require network-wide synchronization (e.g., TDM-based schemes). We assume that the maximum and minimum burst size that can be transmitted on the ring is specified by constants `MaxBurstSize` and `MinBurstSize`, respectively. Furthermore, a transmit queue is not *eligible* for service unless its size is at least equal to the value of `MinBurstSize`.

3.1 Round-Robin with Random Selection (RR/R)

The first protocol a round-robin scheduler at each node to serve the transmit queues, and lets each receiver randomly select a burst from the bursts that arrive simultaneously. We call this protocol *Round-Robin with Random Selection* (RR/R). The operation of the protocol at node i is as follows.

At the transmitting side, the scheduler of node i visits all eligible transmit queues, in a round-robin fashion. If at time t_1, transmit queue j is selected for service, then node i waits for the first control frame that arrives after time t_1. Then, node i writes the burst information and destination address j in its own control slot. After a delay equal to the offset value, node i transmits the burst on its home wavelength.

At the receiving side, when a control frame arrives at node i, it scans the control slots of the control frame, checking for any slot that has i in the destination address field. If more than one such slots are found, node i randomly selects one of them, say k. In this case, all bursts to node i except the one from node k will be lost. Node i then checks whether its receiver is free at the time when the burst from node k arrives at node i, and checks whether its receiver has enough time to tune to another wavelength. If so, it instructs its receiver to tune to node k's home wavelength in order to receive the burst transmission. Otherwise, it gives up on the burst from node k.

3.2 Round-Robin with Persistent Service (RR/P)

The *Round-Robin with Persistent Service* (RR/P) protocol is similar to the RR/R protocol, but it is designed to eliminate receiver conflicts that can be detected prior to the transmission of a burst. The operation of this protocol at node i is as follows.

At the transmitting side, node i maintains a variable `EarliestFreeTime(j)` for each destination node j, which specifies the earliest time at which the receiver of node j would be free. This variable is updated by monitoring the burst information in control slots that have j in the destination address field. The scheduler at node i visits all eligible transmit queues in a round-robin fashion. If at time t_1, transmit queue j is selected for service, then node i waits for the first control frame that arrives after time t_1. Suppose it arrives at time t_2, then node i updates the variable `EarliestFreeTime(j)` based on relevant information in the control frame. Node i also computes the time t_3 that the first bit of its burst would arrive at node j: $t_3 = t_2 + T_i^{(p)} + \text{offset} + \delta_{ij}$, where δ_{ij} is the burst propagation delay from node i to node j. If `EarliestFreeTime(j)` plus the receiver tuning time at node j is less than t_3, then node i writes its burst information in its own control slot, and sends the burst after a delay equal to the offset. If, on the other hand, `EarliestFreeTime(j)` plus the receiver tuning time at node j is greater than t_3, then sending the burst will result in a conflict. In this case, node i does not transmit the burst; instead it waits for the next control frame and repeats the process of transmitting the burst to node j. This is the *persistent* feature of the protocol, in that the round-robin scheduler does not proceed to serve the next transmit queue until the burst to node j has been sent. Note that deferring the transmission of a burst based on a calculation of the earliest free time for receiver j does not altogether eliminate receiver collisions.

At the receiving side, the operation of the protocol is identical to RR/R.

3.3 Round-Robin with Non-persistent Service (RR/NP)

The operation of the *Round-Robin with Non-Persistent Service* (RR/NP) protocol is identical to the operation of the RR/P protocol with one exception. Suppose that at time t_1 node i has selected transmit queue j for service using the RR scheduler. Suppose also that once the first control frame arrives after time t_1, the node determines that transmitting a burst to j would result in a collision. The node refrains from transmitting the burst, and proceeds to serve the next eligible transmit queue upon arrival of the next control frame.

The RR/NP protocol may result in lower delay than RR/P. However, since a node gives up its burst transmission whenever it determines that it will lead to a collision, RR/NP may lead to the starvation of certain transmit queues, and thus, it has fairness problems. Also, RR/NP does not completely eliminate receiver collisions.

3.4 Round-Robin with Tokens (RR/Token)

This protocol uses tokens to resolve receiver collisions. There are N tokens, one for each destination node. A token may be either available or in use. A node can only transmit to a destination node j, if it captures the j-th token. The transmit queues at each node are served in a Round-Robin manner. The operation of the *Round-Robin with Tokens* (RR/Token) protocol is as follows.

At the transmitter side, node i monitors each received control frame. If it finds an available token, it removes it from the control frame, and puts it in its FIFO token queue. Node i also serves the transmit queues in the arrival order of tokens: if the first token in the token queue is token j, node i first checks whether transmit queue j is eligible for service. If not, node i releases token j and proceeds with the next token in the queue. Otherwise, node i constructs the burst to node j, writes the burst information in the next control frame, and sends it after a delay equal to the offset value. Once the burst transmission is complete, node i releases token j to the next control frame. It then proceeds to serve the transmit queue corresponding to the next token in the token queue. Since every node has a FIFO token queue, the order in which tokens circulate around the ring is fixed. Recall that there are only N tokens, one for each destination node. Therefore, transmit queues are served in a Round-Robin manner.

At the receiver side, node i checks each incoming control frame for any control slot indicating a burst transmission to this node. If such a control slot is found, node i instructs its receiver to tune to the appropriate home wavelength for receiving the burst.

RR/Token is a receiver collision free protocol, since there can be at most one burst transmission arriving at a destination node at any time.

4 Numerical Results

We used simulation to compare the protocols described in the previous section. For each of the four protocols RR/R, RR/P, RR/NP, and RR/Token, we consider two variants: one in which the offset calculation is based on ODD, using expression (2), and one in which the offset calculation is based on JET, using expression (1). In our study we consider a ring network with 10 nodes. We set the (electronic) buffer capacity at each node to 10 MBytes. The distance between two successive nodes in the ring is taken to be 5 Km. We assume that the control wavelength runs at 622 Mbps, while each burst wavelength runs at 2.5 Gbps. Each control slot in a control frame is 100 bytes long regardless of the protocol used in the ring. The processing time of a control frame at both the intermediate ($T_i^{(p)}$) and destination nodes ($T_d^{(p)}$) is set to be 10 slot times, and the switch setup time at the destination nodes $T_d^{(s)}$ is 1 μs.

We model the packet arrival process to each node by a modified Interrupted Poisson Process (IPP) [7]. This modified IPP is an ON/OFF process, where both the ON and the OFF periods are exponentially distributed. Packets arrive back to back during the ON period at the rate of 2.5 Gbps. No packets arrive during the OFF period. The packet size is assumed to follow a truncated exponential distribution with an average size of 500 bytes and a maximum size of 5000 bytes. We use the squared coefficient of variation, C^2, of the packet inter-arrival time to measure the burstiness of the arrival process. C^2 is defined as the ratio of the variance of the packet inter-arrival time divided by the squared mean of the packet inter-arrival time. The arrival process is completely characterized by the C^2 and the average arrival packet rate. In all simulations, we set C^2 to 20, and vary the average arrival rate. Packets arriving at a node are assigned a destination node following the uniform distribution.

Effect of Average Arrival Rate. We first investigate the performance of the four protocols when the calculation of the offset is based on ODD. We consider five performance measures: throughput, loss, delay, fairness, and buffer requirement. Since each node is fed with the same arrival process, the average arrival rate we refer to is the average arrival rate to a single node. We set `MaxBurstSize` to 112 Kbytes and `MinBurstSize` to 16 Kbytes.

Figure 4 plots the mean node throughput against the average arrival rate for all four protocols. We observe that RR/Token, a protocol free of receiver collisions, achieves the highest throughput. Among the three protocols in which

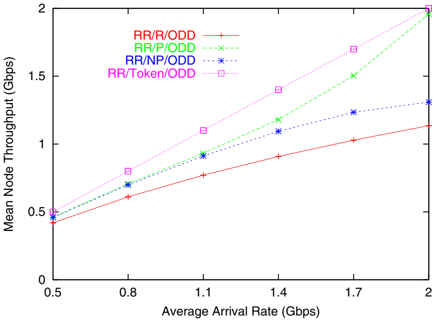


Fig. 4. Mean node throughput

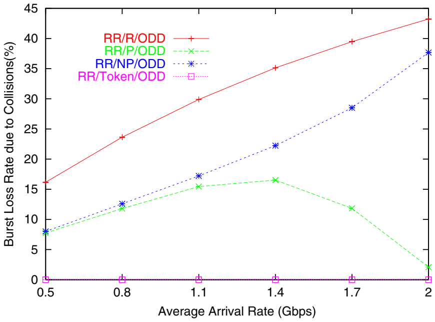


Fig. 5. Burst loss due to collisions

receiver collisions are possible, RR/P achieves the highest throughput, followed by RR/NP and RR/R.

We distinguish between two types of loss. First, packets arriving to find a full buffer at the source node are dropped. In our experiments, we observed that only RR/Token has a 0.01% packet loss rate due to buffer overflow, when the average arrival rate is 2.0 Gbps. The second type of loss occurs when a burst is dropped at the destination due to a receiver collision. Figure 5 plots the burst loss rate due to receiver collisions versus the average arrival rate. As expected, RR/Token never incurs loss due to receiver collisions. For the other three protocols, RR/P has the least burst loss rate, followed by RR/NP and RR/R.

Next, we give an intuitive explanation of the burst loss plots in Figure 5. The behavior of these plots is related to the C^2 of the burst size. If all other parameters are kept the same, a larger burst size C^2 leads to a larger burst loss rate due to receiver collisions. Figure 6 shows the C^2 of the burst sizes as a function of the average arrival rate. We note that the plots in both Figures 5 and 6 have the same pattern. As the average arrival rate increases, the C^2 of the burst size of RR/R and RR/NP increases, and so does the burst loss rate. For RR/P, however, as the average arrival rate increases, the burst size C^2 first increases, it peaks at 1.4 Gbps, and then it decreases. The burst loss rate follows the same pattern. The reason for the change in the C^2 of burst size is that when the burst size reaches a specific point, the `MaxBurstSize` starts to limit the C^2 of burst size.

From the simulation, we also found that the burst loss rate due to receiver collisions of RR/P depends not only on the C^2 of the burst size, but also on another important parameter, the *EnoughData* probability. Recall that in a node, a transmit queue is not eligible for service unless its size is at least equal to the value of `MinBurstSize`. Therefore, when a node turns to serve a transmit queue, the transmit queue may or may not be eligible for service. The probability that a transmit queue is eligible for service when a node turns to serve it is the *EnoughData* probability. We found that for RR/P, an *EnoughData* probability equal to or very close to one leads to a lower burst loss rate due to

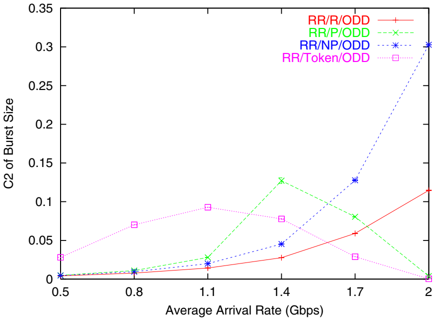


Fig. 6. C^2 of Burst Size

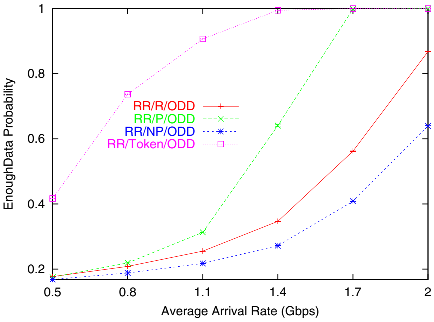


Fig. 7. `EnoughData` probability

receiver collisions than an *EnoughData* probability close to zero. Figure 7 shows the *EnoughData* probability versus the average arrival rate. The *EnoughData* probability of RR/P increases as the average arrival rate increases, reaching 1 when the average arrival rate is 1.7 Gbps.

Figure 8 plots the mean packet delay, including the queueing and propagation delay, versus the average arrival rate. The RR/R protocol has the least delay, followed by RR/NP, RR/P and RR/Token. We observe that, as the average arrival rate increases, the mean packet delay in all protocols first decreases, and then it increases. This behavior is due to the fact that, when the traffic intensity is low, the time for a transmit queue to reach the `MinBurstSize` accounts for the major part of the packet delay. Therefore, as the average arrival rate increases, the time for a transmit queue to reach `MinBurstSize` decreases, which causes the mean packet delay to decrease. The 95% percentile packet delay was also calculated in the simulation. Since the plot trend is the same as that of the mean packet delay, the figure is not shown here.

Let us now compare the four protocols in terms of fairness. We distinguish two types of fairness, namely, throughput fairness and delay fairness. We define the *throughput fairness index of a node i* as

$$\text{Throughput Fairness Index of Node } i \;=\; \left(\sum_{j=1,j\neq i}^{10} (H_{ij} \;-\; \overline{H_i})^2\right) \times \frac{1}{\overline{H_i}^2} \quad (3)$$

where H_{ij} is the throughput from node i to node j, i.e., the average number of bits transmitted by node i and received by node j in a unit time, and $\overline{H_i} = (\sum_{j=1,j\neq i}^{10} H_{ij})/9$. We then define the *throughput fairness index of a protocol* as the average of the throughput fairness indices of all nodes. According to this definition, the smaller the throughput fairness index of a protocol, the better the throughput fairness of the protocol.

Figure 9 shows the throughput fairness index of the four protocols versus the average arrival rate. We observe that RR/R and RR/Token have values very close to zero, meaning that they are throughput fair protocols. We have

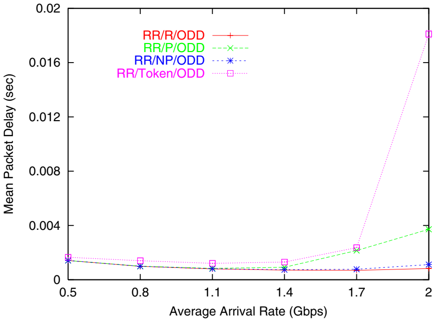


Fig. 8. Mean packet delay

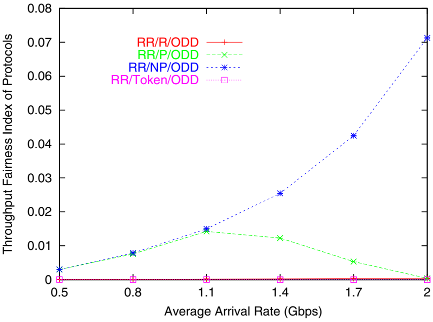


Fig. 9. Throughput fairness index

also computed the throughput from each node to each other node in the ring (these results are not shown here). We have observed that both the RR/NP and RR/P protocols provide better throughput to nodes closer to the source than to nodes far away. This follows directly from the operation of RR/NP and RR/P described in Section 3.3 and 3.2.

The second type of fairness we consider is related to delay. The *delay fairness index of a node i* is defined as

$$\text{Delay Fairness Index of Node } i \;=\; \left(\sum_{j=1,j\neq i}^{10} (W_{ij} \;-\; \overline{W_i})^2\right) \times \frac{1}{\overline{W_i}^2} \qquad (4)$$

where W_{ij} is the mean queueing delay of packets in transmit queue j of node i, and $\overline{W_i} = (\sum_{j=1,j\neq i}^{10} W_{ij})/9$. We also define the *delay fairness index of a protocol* as the average of the delay fairness indices of all nodes. (Note that in defining the fairness index we use the queueing delay only, not the total delay, since the latter includes the propagation delay which depends on the destination node). According to this definition, the smaller the delay fairness index of a protocol, the better the delay fairness of the protocol. Specifically, if the delay fairness index of a protocol is zero, the protocol is perfectly fair since the queueing delay of a packet is insensitive to the source and destination of the packet. For unfair protocols, access to the burst wavelengths may depend on factors such as the relative position of the source and destination nodes in the ring. In this case, some transmit queues may take longer to serve than others, increasing the queueing delay of the respective packets relative to others, and thus, increasing the delay fairness index of the node and protocol.

Figure 10 shows the delay fairness index of the four protocols versus the average arrival rate. We observe that only RR/R has delay fairness index values very close to zero, meaning that it is the only fair protocol in terms of delay. We have also computed the mean packet queueing delay of each transmit at all ring nodes for the four protocols (these results are omitted due to space limitations). We have observed that the RR/NP protocol provides better delay access to wavelengths of nodes far away than to wavelengths of nodes close to the source of a packet, and RR/P and RR/Token do not always provide the best or worst delay access to a specific node.

Overall, the RR/Token protocol achieves the highest mean throughput, followed by the RR/P, RR/NP and RR/R protocols. RR/R has the smallest mean packet delay, followed by RR/NP, RR/P and RR/Token. RR/R also requires the smallest mean buffer requirement, followed by RR/NP, RR/P and RR/Token. The burst loss rate due to receiver collisions for the protocols which are not free of receiver collisions depends on the C^2 of the burst size. The burst loss rate of RR/P also depends on the *EnoughData* probability. Only RR/R is a delay fair protocol, while both RR/R and RR/Token are throughput fair protocols.

Effect of `MaxBurstSize`**.** We also varied the value of `MaxBurstSize` from 32 Kbytes to 112 Kbytes with an increment of 16 Kbytes (not shown here). `MinBurstSize` was set to 16 KBytes, and the average arrival rate to 1.7 Gbps. Simulation results showed that an increase in `MaxBurstSize` leads to an increase in the C^2 of the burst size and to a small change in the *EnoughData* probability; this leads to an increase in the burst loss rate due to receiver collisions, and to a decrease in the throughput of RR/R, RR/NP, and RR/P. However, the decrease in the throughput of RR/R and RR/NP is very small. RR/Token requires a large `MaxBurstSize` so that no packet will be lost due to buffer overflow. Only a very small `MaxBurstSize` could lead to a much longer delay under RR/P and RR/Token. For the other protocols, the change in the mean packet delay because of changes in the `MaxBurstSize` is minimal.

Effect of `MinBurstSize`**.** We varied `MinBurstSize` from 16 Kbytes to 96 Kbytes while keeping the `MaxBurstSize` at 112 KBytes, and the average arrival rate at 1.7 Gbps. Simulation results showed that an increase in `MinBurstSize` leads to a decrease in the C^2 of the burst size and a decrease in the *EnoughData* probability. For RR/R and RR/NP, the decrease in the C^2 of the burst size leads to a small decrease in the burst loss rate due to collisions, which finally leads to a small increase in the mean node throughout. However, for RR/P, a big decrease in the *EnoughData* probability leads to an increase in the burst loss rate due to receiver collisions, which finally leads to a decrease in the mean node throughout. Changes in `MinBurstSize` do not lead to any change in the mean node throughput of RR/Token. Increases in `MinBurstSize` also lead to increases in the mean packet delay of all protocols.

JET vs. ODD. We now focus on the difference between the JET and ODD offset calculations. Due to space limitations, we only consider two protocols: RR/Token and RR/R. Simulation experiments were carried out with the same parameters as above. The results showed that, compared to ODD, JET leads to a longer mean packet delay for all protocols (see Figure 11), which in turn leads to a larger mean buffer requirement, and to a larger packet loss rate due to buffer overflow. Therefore, as a receiver collision free protocol, RR/Token has a lower mean node throughput with JET than with ODD. Moreover, JET naturally leads to delay unfair protocols, but does not change the throughput fairness property of the protocols.

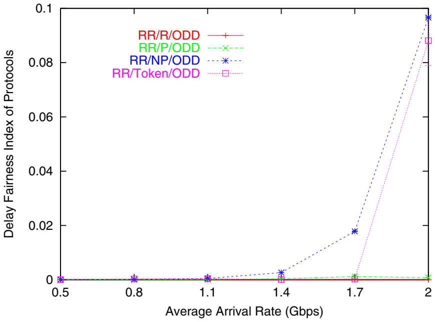


Fig. 10. Delay fairness index

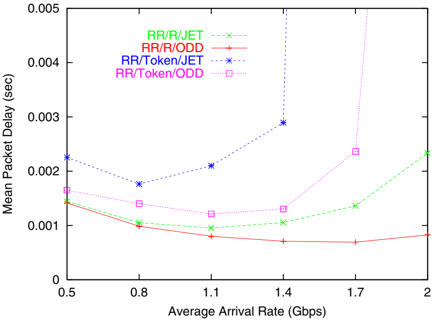


Fig. 11. Mean packet delay

The effect of MaxBurstSize was also investigated. The results showed that all protocols are more sensitive to MaxBurstSize with JET than with ODD. A much larger MaxBurstSize is required in JET than in ODD, in order to get a higher mean node throughput and lower mean packet delay. Results also showed that both ODD and JET are not very sensitive to MinBurstSize. As the MinBurstSize increases, for RR/R, there is no big difference between ODD and JET. But for RR/Token, ODD is always much better than JET in both the mean node throughput and the mean packet delay.

5 Concluding Remarks

We described a WDM metro ring architecture with optical burst switching. Several access protocols were proposed and their performance was analyzed by simulation.

References

1. L. Xu, H. G. Perros, and G. N. Rouskas. Techniques for optical packet switching and optical burst switching. *IEEE Communications*, 39(1):136–142, January 2001.
2. I. Baldine, G. N. Rouskas, H. G. Perros, and D. Stevenson. JumpStart: A just-in-time signaling architecture for WDM burst-switched networks. *IEEE Communications*, 40(2):82–89, February 2002.
3. C. Qiao and M. Yoo. Optical burst switching (OBS)-A new paradigm for an optical Internet. *Journal of High Speed Networks*, 8(1):69–84, January 1999.
4. J. S. Turner. Terabit burst switching. *J. High Speed Networks*, 8(1):3–16, 1999.
5. S. Verma, H. Chaskar, and R. Ravikanth. Optical burst switching: a viable solution for terabit IP backbone. *IEEE Network*, pages 48–53, November/December 2000.
6. Y. Xiong, M. Vandenhoute, and H.C. Cankaya. Control architecture in optical burst-switched WDM networks. *IEEE JSAC*, 18(10):1838–1851, October 2000.
7. W. Fischer and K. Meier-Hellstern. The markov-modulated poisson process (MMPP) cookbook. *Performance Evaluation*, 18:149–171, 1992.

Capacity Efficiency of Distributed Path Restoration Mechanisms in Optical Mesh Networks

Bart Rousseau and Fabrice Poppe

Alcatel, Network Architecture Dept., Network Strategy Group,
Francis Wellesplein 1, B-2018 Antwerpen, Belgium.
Tel.: +32-3-240 70 69, Fax: +32-3-240 48 88
{Bart.Rousseau, Fabrice.Poppe}@alcatel.be

Abstract. In this study the restoration performance of two closely related Sender/Chooser-based distributed path restoration protocols (both extensions of the Self-Healing Network (SHN) to path restoration) is compared. Some pathological situations in which these Sender/Chooser-based restoration algorithms perform suboptimally are identified and, where available, possible solutions are proposed. Built-in mechanisms to resolve contention between Sender/Chooser pairs were found to be very helpful. In addition, the performance of the distributed restoration algorithms studied was found to be close to its theoretical upper bound, suggesting that the pathological situations described may not be that important for real-life networks.

Keywords: distributed path restoration, Sender/Chooser based restoration protocols, optical mesh networks, survivable networks

1 Introduction

Although telecommunications networks have played a crucial role in society for some time now, the last decades society has become increasingly depended on the these networks. As a result the social and economical consequences of network outages have increased dramatically. With typically one cable cut per 0.003/km/year, cable cuts are surprisingly frequent. Given availability requirements of the order of 99.999% or higher (meaning that networks cannot be down for more than 6 min/year on average), it is clear that a fast and reliable recovery strategy is quintessential in contemporary networks.

Since most services can tolerate outages of up to 2 seconds [2], the goal is to develop restoration protocols that can achieve restoration in less than 2 seconds. The time required by a given restoration protocol to restore all failed paths (or as many as possible given the capacity constraints) is termed the restoration speed. It should of course be as low as possible and less than the 2 seconds goal put forward above. A good restoration protocol should combine a high restoration speed with a high restorability (the latter being defined as the percentage of

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 875–886, 2002.

failed paths that can be restored by the protocol under the existing network conditions). In addition the protocol should be capacity efficient and scalable.

Given the advantage of path restoration over link restoration in terms of capacity efficiency, it is surprising to find that relatively little work has been done on path restoration [3,4,5] as compared to link restoration [6,7,8,9,10,11,12,13, 14,15]. One promising approach for path restoration is the extension of the Self-Healing Network (SHN) protocol [12] to path restoration introduced by Iraschko and Grover [4]. The SHN and its extensions form a class of Sender/Chooser-based restoration protocols. In this study the restoration performance of two extensions of the SHN to path restoration is compared. As a result of this study some potential weaknesses of these protocols were identified. These will be illustrated in the first part of this work and, where available, possible solutions will be proposed. In the second part, our implementations of the protocol are used to verify what the consequences are of these issues in tightly dimensioned real-life networks. In the closing section conclusions are drawn.

2 Sender/Chooser Based Path Restoration Protocols

As already mentioned in the introduction, there exists a class of distributed restoration protocols that have as common characteristic the use of a Sender/Chooser mechanism. In this section a brief overview of the Sender/Chooser mechanism as applied to path restoration is given. Two variants of the basic protocol will be described. The first can be considered a direct extension of the SHN to path restoration. The second, introduced in ref. [4], is a modification in which an interference number is used as a measure to avoid contention for available resources. Only a brief description of these protocols will be given here, for more details the reader is referred to the original literature [2,4,12].

It is assumed that in case of a link failure, both the origin and destination nodes of all paths using the affected link are notified. When the terminating nodes of a path receive this notification, Sender/Chooser arbitration occurs, *i.e.* one node takes on the Sender role while the other takes on the Chooser role. This is done independently by both nodes using some arbitrary rule that yields a uniquely defined result (*e.g.* based on the ordinal number of the nodes). At this point the actual restoration process can start.

For each failed path for which a given node acts as a Sender, it sends a forward flooding (FF) message over each available outgoing link, thereby temporarily reserving spare capacity. These messages then propagate to the adjacent nodes. Such nodes that are neither Sender nor Chooser for a given path are called Tandem nodes (Note that a node can simultaneously be a Sender for one path, a Tandem node for another path and a Chooser node for yet another path. For any given path however any node is either a Sender, a Tandem or a Chooser node). The Tandem node rules discussed below, manage the contention amongst incoming messages for subsequent retransmission on the spare channels available at that node. Messages initiated from the node in its Sender role are integrated into the same overall competition. When a message reaches the Chooser node, this node answers with a reverse linking (RL) message. This messages traces back to the Sender node, meanwhile locking the required channels and releasing resources no longer required. The restoration path is locked and reserved once the RL message reaches the Sender node.

As mentioned above, for each failed path for which a given node acts as a Sender, it creates an internal arrival message for each available outgoing link. Each of these messages is assigned a locally unique index value. This permits tracebility in reverse linking to a specific port back at the Sender. These internal arrival message together with the external arriving messages (messages arriving over an incoming channel) compete for the available outgoing channels in order to get forwarded over one of these channels. In order to decide which messages are allowed to propagate and which are not, some kind of priority value is needed. One possibility is to use the repeat value (also called hop count) in order to discriminate between messages (this could be considered to be a direct extension of the Self-Healing Network to path restoration). In ref. [4] an interference number (IN) is introduced in order to avoid creating paths that could render a large number of other restoration paths infeasible. (A precise definition of the interference number will be given below). The authors propose to use this IN to decide which messages to forward.

Of each (Origin, Destination, Index) family (see ref. [4]), only one message, that with the lowest IN, is considered as a candidate for propagation. All the messages that are candidates (termed ‘precursors’ in ref. [4]) are then arranged in a table in order of increasing IN (breaking ties using the repeat count and then the index number if needed). Since the internal arrival messages, by construction, have an IN equal to zero, these will be at the top of the table. This table, in combination with the available capacity (number of outgoing channels) on each outgoing link, determines which messages will be propagated and which messages will be stopped in the current node. Next the table is examined from top to bottom, propagating each message over as many available outgoing links as possible using only one channel per message on any link (Note that although a given message may not be forwarded (or only partially) because the required capacity is not available, other messages further down the table might still be able to propagate). Messages are not forwarded over the link over which they arrived.

Some messages will not be able to propagate over all possible links. The number of messages that would have been forwarded on a link given enough capacity minus the number of messages that can be forwarded given the actually available capacity is called the interference number of that link (*i.e.* the number of messages ‘blocked’ by the messages that got through). When a message is forwarded over a link its IN is increased by the IN of that link.

3 Potential Shortcomings of Distributed Sender/Chooser-Based Path Restoration Algorithms

While working on our implementation of the protocol described in ref. [4] we identified a number of situations in which the Sender/Chooser-based restoration algorithms perform suboptimally, *i.e.* cases in which the final restoration ratios are lower than those achievable by centralized restoration algorithms. These will be discussed below and, where available, possible solutions are proposed.

3.1 Dependency on Topology

In ref. [4] reference is made to the so-called trap topology (see Fig. 1) to illustrate that a shortest path approach to restoration can in some cases lead to less optimal results than a maximum flow based approach. As an example suppose that two paths have been lost between nodes A and D. In this case, using a shortest path algorithm yields only one path (A-B-C-D) whereas using a maximum flow algorithm yields two paths (A-E-C-D and A-B-F-D). Neither using the repeat value nor using the IN in the protocol ensures that the two possible paths are obtained. This can be seen as follows. At a given moment in time, two messages will be received in Node C (one via A-E-C and one via A-B-C). Since these messages have identical IN and repeat values (an IN and repeat count of 1 and 2, respectively) the decision of which message to forward will only depend on the (arbitrarily assigned) index values of the messages. Depending on the relative magnitude of these index values either one or two paths will be obtained. In this case the restoration ratio is thus dependent on the arbitrary assignment of index values to the messages leaving the Sender node. Although this is a somewhat unsatisfactory situation, it is not immediately clear how this can be avoided. Note that this problem can occur if there is only one Sender/Chooser pair (such as *e.g.* in link restoration). Its occurrence simply depends on topological details and is unrelated to contention between restoration pairs. As such it is not specific to path restoration.

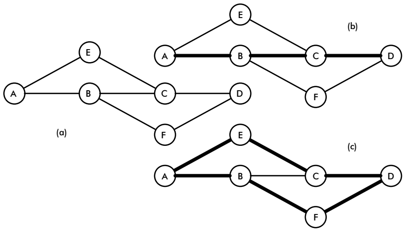


Fig. 1. Trap topology. Each link is assumed to have one spare channel available for restoration. Using a shortest path approach yields one path (*b*) whereas a maximum flow approach yields two paths (*c*). As explained in the text, which of these two possibilities is obtained using the protocols described above, depends on the arbitrary assignment of index values to the messages sent from the Sender node.

3.2 Unfairness towards FF Messages Close to the Chooser Node

The network illustrated in Fig. 2 is used in ref. [4] to illustrate the need for a protocol for path restoration that has some built-in means to avoid contention. Suppose that, in the network given in Fig. 2, each link in the network has a spare capacity of one channel. Further assume that we have lost 3 paths between nodes 1 and 8 and 4 paths between nodes 2 and 5. Nodes 1 and 2 are chosen to be the Sender nodes and nodes 5 and 8 the Chooser nodes.

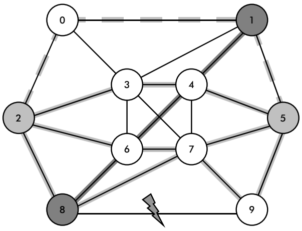


Fig. 2. Of the five possible paths only four are found due to the priority given to internal arrivals (see text for details).

The optimal solution in this case contains five paths (1-4-6-8, 2-0-1-5, 2-3-4-5, 2-6-7-5, and 2-8-7-9-5). However, using the protocol described above, only four paths (1-4-6-8, 2-3-4-5, 2-6-7-5, and 2-8-7-9-5) will be obtained. The reason that the path 2-0-1-5 is not found can be understood as follows. Node 1 is Sender node for four paths and as such will try to propagate four FF messages over each of its links (but due to the capacity constraints can send only one). When at a given moment in time a FF message sent by Node 2 arrives in Node 1 via node 0, it will be stopped there because it has a higher repeat value than the internal arrival messages of node 1 (that by definition have a repeat count of zero). Introducing an IN does not improve the situation since the internal arrival messages have, by construction, an IN that is not higher than that of any other message. In addition, as before, it will always have a lower repeat value. The fact that not all possible paths are obtained is thus closely coupled to the precedence that is given to the internal arrivals in the Tandem node rules. Note that this problem is specific for situations in which there are several Sender/Chooser pairs simultaneously performing restoration (regardless whether the repeat count or the IN is used).

3.3 Deadlocks

The problem mentioned in the previous subsection is a specific case of the more general phenomenon that under certain conditions a deadlock can occur. Such a case is illustrated in Fig. 3. Because messages have a higher probability to propagate near their own Sender node (due to their lower repeat count) than other messages, they risk blocking other paths. In some circumstances (especially when a very limited amount of spare capacity is available) this can cause the restoration procedure to stop although a (large) number of paths is still feasible given the available resources. Although usage of the IN can in some cases alleviate this problem somewhat, it cannot be excluded completely.

3.4 Race Conditions in the RL Procedure

In ref. [4] it is argued that it is preferable not to give reverse linking (RL) absolute priority, *i.e.* to allow the RL procedure to legitimately fail en-route. However, if no special measures are taken (requiring, as will be shown below, an extension

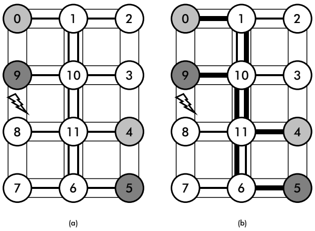


Fig. 3. Illustration of a deadlock situation in which none of the possible paths are found. Messages from Sender node 5 are stopped in node 10 because they have a larger repeat value than those emitted by node 0. For the same reason, messages from Sender node 0 are stopped in node 11.

of the protocol) this can lead to erroneous results in which a restoration path is believed to exists, where in fact only a part of this path is available. This can be illustrated using the following example (see Fig. 4).

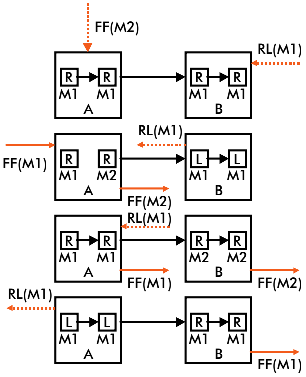
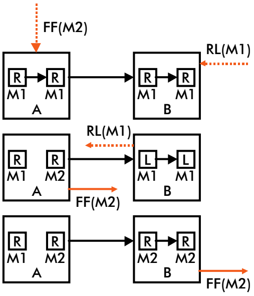



Fig. 4. Successful (*left*) and erroneous (*right*) cancellation of a RL procedure (see text).

The initial situation is such that a message M1 has traced a path from its Sender node to its Chooser node through Tandem nodes A and B. Now suppose that before the RL message corresponding to M1 arrives back at node A, the following occurs. The port over which M1 was sent from node A to node B is taken over by another message (M2). Then, before M2 arrives at node B, M1 reclaims the port (because *e.g.* the precursor of M2 disappears). This means that when the RL message corresponding to M1 arrives back at node A, it will be propagated in the direction of the Sender node, because a mach between a FF and an RL message was found. When this RL message arrives at the Sender node, a complete locked-in restoration path is supposed to exist between the Sender and Chooser nodes. In reality however, only a partial path exists since the message M2 will have torn down the part of the path closest to the Chooser node (*i.e.* between node B and the Chooser node).

To solve the problem mentioned above, we need to be able to ensure that the port has not been taken over at any time before the RL message arrives. For this purpose a locally unique identifier should be assigned to each message when passing through a port. This enables the RL message later on to verify that the port was not taken over by another message between the time the that the original FF message left this port and the time the RL message arrives in this port. In our current implementation a time stamp is used for this purpose, but any locally unique identifier (per port) will do.

3.5 Hold-off Time

Suppose that in the network given in Fig. 5 two paths have been lost, one between nodes 5 and 14, and one between nodes 4 and 6. Given a spare capacity distribution as shown in Fig. 5 both paths can in principle be restored. Using only the repeat value in the protocol described above, only one path will be obtained (5-9-10-14). One may be tempted to think that using the IN will result in two paths. That this is not the case can be understood as follows. Lets call the path from node 5 to node 14 P1 and that from node 4 to node 6 P2. The messages involved in the restoration of these paths can then be identified as M1 and M2 for paths P1 and P2, respectively. The first event that occurs is the arrival of M1 in node 9. Since it is the only message present at that time in this node for forwarding on the link from node 9 to node 10 and one spare channel is available on this link, the message will be forwarded over this link with an IN equal to zero. Shortly after, M1 arrives in node 9. We now have two messages to forward over the link 9-10 and only one channel available for restoration. This means that only one message will be forwarded and that it will have its IN incremented by 1. Since both messages arrive with an IN of zero, but M1 has a lower repeat count (1 for M1 versus 2 for M2) M1 will be propagated over link 9-10 with an IN of 1. As a consequence, node 1 first receives M1 (via 5-9-10-14) with an IN of zero. Node 14 reacts by sending an RL message in the direction of node 5 (via the link 14-10). However, before the RL attempt can succeed, the path is taken over by M1 with an IN of 2. When this message arrives in node 14 (again via 5-9-10-14), a new RL attempt will be started. Note, that

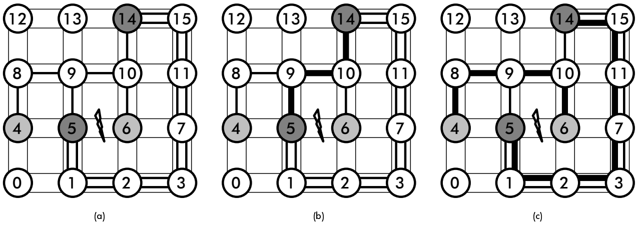


Fig. 5. Using just the repeat value, only a single path is found (*a*). Using the IN, either one (*b*) or two (*c*) paths are found, depending on whether or not a hold-off time is used.

at the same time, a message M1' is tracing an alternative restoration path for P1 (5-1-2-3-7-11-15-14). However, since node 14, by the time M1' arrives, has already responded with an RL event to M1, it will ignore the arrival of M1' even though this message has a lower IN (1 for M1 and 0 for M1'). Therefore, when the RL event, started upon the arrival of M1 in node 14, succeeds, only one path is found (5-9-10-14).

This can be solved by introducing a hold-off time before which no RL event is allowed to start. Two scenarios are possible. Either the hold-off time is applied after the reception of the alarm signal or it is applied after the arrival of the first FF message for the given path (Note that this is somewhat similar to the use of a hold-off time in the FITNESS algorithm [15,16]). The result of introducing a hold-off time is that (at least when the hold-off time is chosen sufficiently long) node 14 will not start any RL attempt until both M1 and M1' have arrived. Since M1' has a lower IN than M1, node 14 will use M1' for RL. When this RL message sent from node 14 arrives back in node 5, a restoration path for P1 is found (5-1-2-3-7-11-15-14). In addition, upon the arrival of the RL message in node 5, all remaining messages for P1 will be canceled. As a consequence link 9-10 will become available for restoration of P2. Thus, introducing a hold-off time ensures that the IN can fully play its role. As a net result in this case, two paths instead of only one are obtained.

3.6 Dynamicity Due to IN

One of the major consequences of introducing the interference number is the increased 'dynamicity' of the protocol. The repeat count of a message that has arrived in a given node through a given path does not change in time. The IN however can change due to events taking place in any of the nodes the message passed through on its way to its current location (in fact the IN of the message does not change but more recent messages for the same path but with a different IN will take over the ports in use by the original message). This increased dynamic behavior can in some circumstances (especially in larger networks with a small amount of spare capacity) dramatically slow down the restoration process (in some cases even to the extent that no restoration paths are found within the two seconds time limit). We have therefor decided in our implementation of the protocol to reduce this 'dynamicity' as follows. Suppose that a message has traced a path from its Sender node in the direction of its Chooser node via Tandem node A. In our implementation, when the broadcast pattern in node A changes (because *e.g.* a new message has arrived) the old message is not replaced by a new message with an updated IN. This is a different approach than that used in the original description of the protocol [4]. Letting messages that have already been sent keep their INs, even when they should have been updated due to changes in the local forwarding pattern, reduces the dynamic behavior described above. However, at the same time, this limits the extent to which the IN can fulfill its role of avoiding the creation of unfavorable paths.

4 Simulation Results

In this part the results obtained using our implementation of the protocol described above will be discussed. The performance of this algorithm in tightly provisioned networks is studied. The results indicate that the phenomena described in the previous part are not that important for real-life networks.

4.1 Parameters

Messages with a size of 512 bits are sent over a dedicated signaling channel having a bit rate of 152 Mbps. The propagation speed on links was set to 2 10^5 km/s. A processing delay of 0.5 ms was used incremented with 1ms when the tandem logic required reevaluation of the composite forwarding pattern.

4.2 Networks

In the simulations three networks (with 13, 19, and 30 nodes, respectively) were used in combination with two demand patterns for each network (see Table 1 for details). These networks, of which one is depicted in Fig. 6, have a relatively high nodal degree and are therefore representative of core networks. The networks only have the minimum amount of spare capacity required to allow for 100% restorability for all single link failures in a revertive restoration scenario, *i.e.* without release of the capacity used by the affected lightpaths prior to restoration (stub release). The method that was used to place spare capacity was described in ref. [17]. It does not take into account the details of the distributed restoration algorithm. As such, the networks studied can be used as benchmarks: because of the spare capacity placement, optimized centralized restoration mechanisms can achieve a restoration ratio of 100% for every single link failure. Through simulations we studied how close to this upper bound the performance of the distributed restoration mechanisms is.

Table 1. Number of nodes (N), number of links (L), average nodal degree (n), average span length in km (l), number of origin/destination pairs ($N_{O/D}$), total demand for all origin/destination pairs (D), average number of working channels per link (W), average number of spare channels per link (S) and network redundancy (R) for the different networks used in this study.

	N	L	n	l	$N_{O/D}$	D	W	S	R
I13S	13	24	3.7	105	31	57	4.2	3.5	0.85
I13L	13	24	3.7	105	67	209	16.7	10.3	0.62
E19S	19	40	4.2	779	78	132	6.3	2.9	0.46
E19L	19	40	4.2	779	110	246	11.7	4.8	0.41
I30S	30	59	3.9	146	114	261	10.5	6.4	0.61
I30L	30	59	3.9	146	301	972	45.7	23.5	0.51

4.3 Results and Discussion

The restoration ratio and restoration time obtained for all possible single-link failures in the networks described above are summarized in Table 2, both with (non-revertive) and without (revertive) stub release. Note that, despite the tightly optimized amount of spare capacity, high restoration ratios are obtained, both using the repeat count and using the IN. For the non-revertive approach, in all cases where less than 100% restorability was achieved, this was due to the path sets chosen by the algorithm and not due to *e.g.* deadlocks such as described above. In the non-revertive case using the interference number has a negligible effect. In the revertive approach, however, use of the interference number clearly yields better restoration ratios, *e.g.* for the 30-node network, 86% of the 5390 lost paths could be restored using the repeat count, whereas using the INs this increased to 94%. Note that although for all the networks the spare capacity was dimensioned to allow for complete recovery from all single-link failures using a revertive approach, a restorability of 100% was not obtained in all cases using the distributed restoration protocol.

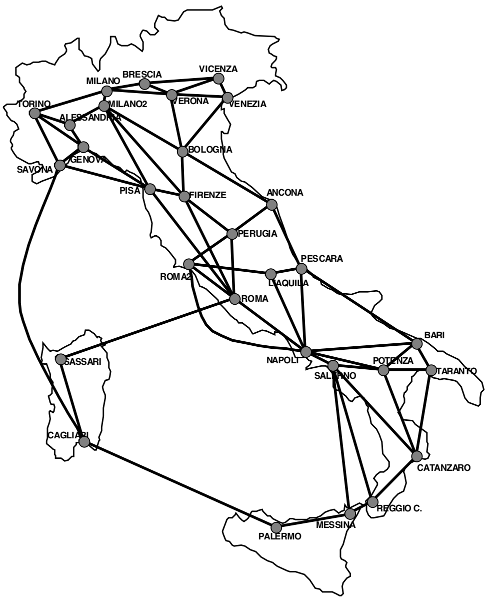


Fig. 6. One of the networks used in this study. The network characteristics are summarized in Table 1.

In addition to the single link failure simulations discussed above some double link failures where performed using the 13-node network. Also in this case does using the INs yield improved results compared to using only the repeat count. Of the 2820 failed paths only 2014 could be restored using the repeat count whereas

Table 2. Number of restored paths ($N_{restored}$), number of failed paths (N_{failed}), time at which the last restoration paths was found (t_{max}, in ms), average restoration time (t_{avg}, in ms) and amount of spare capacity used for restoration paths (SC_{used}).

		No stub release	*Stub release*			
	N_{failed}	$N_{restored}$	$N_{restored}$	t_{max}	t_{avg}	SC_{used}
IN						
I13S	200	200	200	39	30	16
I13L	800	800	800	61	39	24
E19S	500	498	500	133	67	21
E19L	938	935	938	162	73	22
I30S	1236	1223	1230	77	44	16
I30L	5390	5078	5388	126	56	18
Repeat count						
I13S	200	200	200	37	30	15
I13L	800	800	800	57	38	23
E19S	500	498	498	118	60	19
E19L	938	931	938	144	64	21
I30S	1236	1208	1235	67	39	15
I30L	5390	4623	5390	115	53	18

2080 could be found using the interference numbers (note that the network was only designed to handle single link failures).

5 Conclusions

In this study the restoration performance of two closely related distributed path restoration protocols, both extensions of the SHN, is compared. The first can be considered to be a direct extension of the SHN to path restoration. The second, based on an interference heuristic, has a built-in measure to avoid contention for spare resources. In the fist part a number of situations were identified in which Sender/Chooser-based restoration protocols clearly will perform suboptimally and, where available, solutions were proposed. In the second part the restoration behavior of the two restoration protocols is examined using simulation on tightly dimensioned real-life networks. Built-in mechanisms to resolve contention between Sender/Chooser pairs were found to be very helpful. In addition it was found that the restoration performance of the distributed restoration algorithms studied was close to its theoretical upper bound, suggesting that the pathological situations we discussed may not be that important for real-life networks.

References

1. Doucette, J., Grover, W.D.: Influence of Modularity and Economy-of-Scale Effects on Design of Mesh-Restorable DWDM Networks. IEEE Journal on Selected Areas in Communications **18** (2000) 1912–1923

2. Grover, W.D.: Distributed Restoration of the Transport Network. In: Aidarous, S., Plevyak, T. (eds.): Telecommunications Network Management into the 21st Century. IEEE Press (1994) Chapter 11
3. Doshi, B. T., Dravida, S., Harshavardhana, P., Hauser, O., Wang, Y.: Optical Network Design and Restoration. Bell Labs Technical Journal January-March (1999) 58–84
4. Iraschko, R.R., Grover, W.D.: A Highly Efficient Path-Restoration Protocol for Management of Optical Network Transport Integrity. IEEE Journal on Selected Areas in Communications 18 (2000) 779–794
5. Wuttisittikulkij, L., O'Mahony, M.J.: Use of Spare Wavelengths for Traffic Restoration in a Multiwavelength Transport Network. Fiber and Integrated Optics **16** (1997) 343–354
6. Chow, C.E., Bicknell, J.D., Mccaughey, S.: Performance analysis of fast distributed link restoration algorithms. Int. J. Commun. Syst. **8** (1995) 325–345
7. Chujo, T., Komine, H., Miyazaki, K., Ogura, T., Soejima, T.: Distributed self-healing network and its optimum spare capacity assignment algorithm. Electron. Commun. Japan. Part 1. **74** (1991) 1–8
8. Doverspike, R., Sahin, G., Strand, J., Tkach, R.: Fast Restoration in a Mesh Network of Optical Cross-connects. Proceedings of the Optical Fiber Communications Conference, OFC '99 (1999)
9. Ellinas, G., Hailemariam, A.G., Stern, T.E.: Protection Cycles in Mesh WDM Networks. IEEE Journal on Selected Areas in Comm. **18** (2000) 1924–1937
10. Lee, H., Song H.-G., Chung, J.-B., Chung, S.-J.: Preplanned rerouting optimization and dynammic path rerouting for ATM VP restoration. Telecomm. Syst. **14** (2000) 243–267
11. Gersht, A., Kheradpir A., Shulman A.: Dynamic Bandwidth-Allocation and Path-Restoration in SONET Self-Healing Networks. IEEE Transactions on Reliability **45** (1996) 321–331
12. Grover, W.D.: Selfhealing networks — A distributed algorithm for k-shortest link-disjoint paths in a multi-graph with application in real-time network restoration. Ph.D.-thesis. University of Alberta (1989)
13. Poppe, F., De Neve, H., Petit, G.H.: Constrained Shortest Path First Algorithm for Lambda-Switched Mesh Optical Networks with Logical Overlay Och/SP Rings. 2001 IEEE Workshop on High Performance Switching and Routing (2001) 150–154
14. Sakauchi, H., Nishimura, Y., Hasegawa, S.: A self-healing network with an economical spare-channel assignment. Proc. IEEE Globecom'90 (1990) 438–443
15. Yang, C. H., Hasegawa, S.: FITNESS: Failure Immunization Technology for Network Service Survivability. Proc. IEEE Global Telecomm. Conf., Globecom '88 (1988) 1549–1554
16. Bicknell, J., Chow, C.E., Seyd, S.: Performance Analysis of Fast Distributed Network Restoration Algorithms. Proceedings IEEE GLOBECOM '93 (1993) 1596–1600
17. Poppe, F., Demeester, P.: Economic Allocation of Spare Capacity in Mesh-Restorable Networks. Proc. Of the Sixth International Conference on Telecommunication Systems, Modeling and Analysis (1998) 77–86

Helios: A Broadcast Optical Architecture*

Ilia Baldine[1], Laura E. Jackson[1], and George N. Rouskas[2]

[1] MCNC ANR, Research Triangle Park, NC, USA, ibaldin,lojack@anr.mcnc.org
[2] Department of Computer Science, North Carolina State University, NC, USA,
rouskas@eos.ncsu.edu

Abstract. In this article we present a new all-optical broadcast LAN architecture and an accompanying signaling protocol. The distinguishing characteristics of this architecture are its fault-tolerant design and its collision-free nature, which allows it to achieve high throughput in a broadcast environment. The flexibility of the design allows different schedulers to be used, which can introduce new features into the network (e.g. multicast and QoS) as well as optimize its behavior for the specific setting in which it is used.

1 Introduction

Wavelength division multiplexing (WDM) optical networks are a viable technology for a next-generation network infrastructure that supports a diverse set of existing, emerging, and future applications [8]. WDM bridges the gap between lower electronic switching speeds and ultra high optical transmission speeds. Dividing the enormous information carrying capacity of single mode fiber into a number of channels, each on a different wavelength and operating at peak electronic speed, WDM makes it possible to deliver aggregate throughput on the order of Terabits per second. WDM technology initially was deployed in point-to-point links and has also been extensively studied, theoretically and experimentally, in wide area or metropolitan area distances [7]. Several WDM local area testbeds have also been implemented [5] or are currently under development [6, 1].

In this article we present `Helios` – a WDM all-optical architecture for a local area network and an accompanying signaling protocol. The packet-oriented `Helios` architecture enjoys independence of the number of nodes and the number of supported wavelengths, and relies on scheduled access to the medium, guaranteeing higher utilization. `Helios` is part of a DARPA-funded project aimed at demonstrating the feasibility and potential of optical access networks. This effort is a logical continuation of earlier work performed at NCSU ([11], [12]). Following an overview of the architecture in Sect. 2, we describe the signaling protocol in Sect. 3 and the basic `Helios` scheduling algorithm in Sect. 4.1. We conclude in Sect. 5.

* Supported by DARPA under Contract No. F-30602-00-C-0034 and NSF under Grant No. 9701113.

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 887–898, 2002.

2 The Helios Architecture

The Helios network employs a passive star coupler (PSC) as a broadcast medium to connect all nodes in the network, making Helios a *single-hop* WDM network. The entire path between source and destination in such a network is entirely optical; no electro-optic conversion of the signal is necessary [9]. Helios uses a smaller number of wavelengths than the potentially large number of nodes. The Layer 3 protocol could be either IPv4 or IPv6.

Communication in a Helios network is made collision-free by a non-preemptive gated scheduling protocol. A single *master* node in the network calculates and disseminates the schedule, while other nodes use this schedule to time the transmission of data to their peers. There are two types of nodes: *candidate* nodes, which are eligible to serve as the master node should the current master node fail, and *slave* nodes, which are not. Such a distinction is necessary because a network will likely be composed of servers and workstations, where the workstations lack the necessary computing resources to perform the master node's duties. Furthermore, workstations may allow low priority user access, making them vulnerable to security attacks that could disrupt the network.

The Helios network utilizes a Fast Tunable Transmitter – Slowly Tunable Receiver (FTT–STR) approach, where *fast* implies low to sub-microsecond tuning times and *slow* implies hundreds of microseconds to tens of milliseconds. For packet transmission and scheduling purposes, the lasers are considered tunable and the receivers fixed. However, in order to balance the load in the network, the receivers may be retuned from time to time, on the order of seconds.

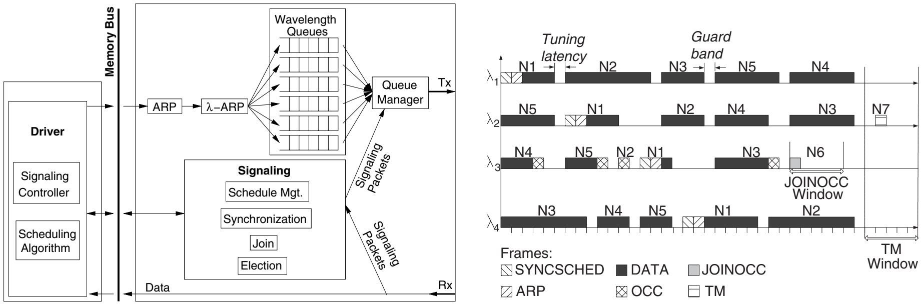
Helios differs from other WDM networks currently under development in several respects: it operates within a broadcast-and-select environment, it is collision-free, and it is packet-switched instead of circuit-switched. At the same time, the Helios architecture provides for such important LAN features as native QoS support and multicast, described in [4] and [13].

2.1 High Level Node Design

Figure 1(a) highlights the various hardware, software, and firmware components of a Helios network adapter. The software **Driver** module consists of two sub-modules. The **Signaling Controller** coordinates the operation of all other software and hardware modules. The **Scheduling Algorithm** calculates new schedules based on queue occupancies provided by all nodes in the network; it is called infrequently, in response to changes in the traffic pattern or simply periodically.

In hardware, the **Signaling** module of the adapter contains four sub-modules: **Schedule Management** forms and processes frames related to scheduling, **Synchronization** enables all communication to occur in hard real time, **Join** allows a new node to join a Helios network, and **Election** manages the selection of a new master node when the current master node fails.

The **ARP** and **λ-ARP** tables enable a Helios node to perform IP-to-MAC address resolution and MAC-to-receive-wavelength resolution, respectively. The



(a) High level node architecture

(b) Example superframe

Fig. 1. Overview of `Helios` node architecture and superframe organization

master node keeps track of ARP and λ-ARP mappings and distributes them via ARP frames to all other nodes. Outgoing IP packets are buffered in the **Wavelength Queues** on a per-wavelength basis prior to transmission. The **Queue Manager** serves the wavelength queues in FIFO order and controls which frames are transmitted.

2.2 Frames and Superframes

The time required to complete the transmissions of one full schedule in `Helios` is a *superframe*. A superframe further consists of frames, continuous sequences of octets transmitted by nodes on individual wavelengths; Table 1 shows the different frame types. `Helios` uses non-preemptive schedules, thus within each superframe a node transmits on a particular wavelength at most once.

Table 1. `Helios` frame types and their function

DATA	Carries regular data
MDATA	Carries multicast data
TM	Measures roundtrip delay to PSC
OCC	Transmits queue occupancies to master node (Routine mode)
JOINOCC	Transmits queue occupancies to master node (Join mode)
SYNCSCHED	Carries scheduling information
ARP	Carries MAC to wavelength index mapping (λARP)
OAM	Carries error and management information about network state
AVAIL	Announces availability of a candidate node to become the master node during scheduler election

The master node calculates the schedule based on other nodes' packet queue occupancies, which it learns through the OCC frames sent by other nodes during routine network operation. Once calculated, the schedule is then broadcast on each wavelength inside the SYNCSCHED frame, which the master node transmits on every wavelength every superframe. A schedule contains **windows**, or intervals of time, during which a particular node may transmit a frame.

Figure 1(b) shows the position of various frames and windows within a superframe. In this example, N1 is the master node and its receive wavelength is $\lambda 3$. There is a JOINOCC window on $\lambda 3$ (with a JOINOCC frame in it), and there is an attached TM window at the end of the superframe. Two nodes are in different stages of joining the network: N6 is sending a JOINOCC frame containing its queue occupancy information to the master node so that it can be included in the next schedule. Meanwhile, N7 is performing Time Measurement; its TM frame can be seen inside the TM window. Time measurement is the first operation a new node must perform when joining the network, in order to synchronize frame reception and transmission.

3 Network Operation: The Helios Signaling Protocol

The operation of a node in the Helios network is separated into the six different modes shown in Table 2. Following an overview of each mode, we discuss one, Routine Mode, in detail. When the network comes up after having been completely powered down, no master node has yet been designated, no frames are traveling, and no synchronization information is available. The first task is the election of a master node; candidate nodes enter **Election Mode** while slave nodes sleep. The operation of Election Mode assumes that candidate nodes are equipped with slowly tunable receivers; otherwise, a network administrator must designate the master node.

Once a master node has been elected, it circulates scheduling and synchronization information in SYNCSCHED frames. Now other nodes may join the network, by proceeding through the **Time Measurement** and **Join** modes. In Time Measurement, a node calculates its psc_offset, the propagation delay to the PSC. All times are measured locally, and the transmissions are done in relation to the PSC time. Since collisions can occur only at the PSC, each node uses its psc_offset to ensure that its transmissions reach the PSC at the exact

Table 2. Modes of operation in the Helios network

Time Measurement	a new node measures its propagation delay to the PSC
Join	a new node contacts the master node with its bandwidth requirements
Election	a candidate node participates in the election of a new master node
Routine	a node transmits and receives data and related signaling frames
Scheduling	same functions as routine, plus must create and distribute new schedules
Error	error detection, report and recovery

time prescribed by the schedule. After Time Measurement, a node enters **Join Mode**. It informs the master node of its traffic demands via the JOINOCC frame; the master node then calculates a new schedule to include this new demand. The joining node waits to hear the new schedule before beginning normal transmissions.

It is possible for a collision to occur when two or more nodes attempt to join a `Helios` network at the same time. Two nodes assigned to the same listening wavelength could experience a collision during Time Measurement, or two nodes may transmit a JOINOCC frame to the master node during the same JOINOCC window. The protocol includes backoff algorithms to resolve such contention.

After **Join**, a node enters **Routine Mode**, where it remains unless an error condition occurs. The receive hardware extracts the schedule from arriving SYNCSCHED frames and forwards incoming data frames to the driver; meanwhile, the transmit hardware transmits control frames and data frames from its wavelength queues onto the appropriate outgoing wavelengths, according to the current schedule. Among these transmissions is an OCC frame sent to the master node, once per superframe, to communicate the node's packet queue occupancies; the master node uses queue occupancies from all nodes to recalculate the schedule. Unlike Time Measurement and Join modes, Routine Mode is collision-free. The `psc_offset`, first measured during Time Measurement, is also measured periodically during Routine Mode, in a collision-free manner.

3.1 Routine Mode: The Receiver State Machine `>routine<`

Figure 2 shows the state machine `>routine<` which governs the receiver's actions during routine mode. A software signal to `>routine<` initiates the transition out of IDLE and into the ROUTINE LISTEN state. When a SYNCSCHED frame arrives, `>routine<` first checks whether its own MAC address (`my_node_ID`) is included in the schedule. If the node has been left out of the schedule, `>routine<` sends the "NOT_IN_SCHED" signal to the Signaling Controller and returns to IDLE. The Signaling Controller then exits Routine Mode and moves to Error mode.

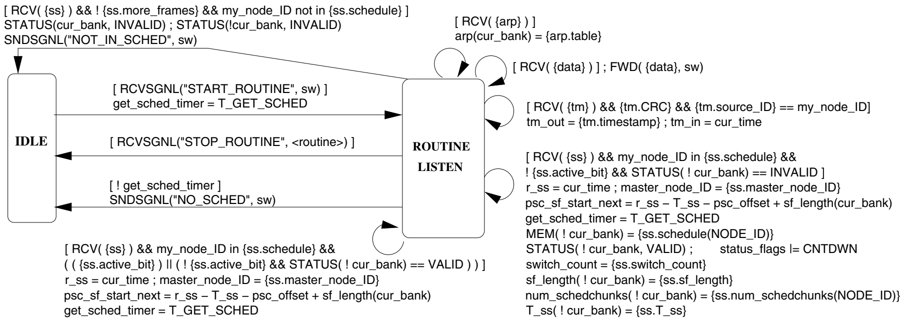

[RCV({ss}) && ! {ss.more_frames} && my_node_ID not in {ss.schedule}]
STATUS(cur_bank, INVALID) ; STATUS(!cur_bank, INVALID)
SNDSGNL("NOT_IN_SCHED", sw)
[RCV({arp})]
arp(cur_bank) = {arp.table}
[RCV({data})] ; FWD({data}, sw)
[RCVSGNL("START_ROUTINE", sw)]
get_sched_timer = T_GET_SCHED
IDLE
[RCVSGNL("STOP_ROUTINE", <routine>)]
ROUTINE
LISTEN
[! get_sched_timer]
SNDSGNL("NO_SCHED", sw)
[RCV({tm}) && {tm.CRC} && {tm.source_ID} == my_node_ID]
tm_out = {tm.timestamp} ; tm_in = cur_time
[RCV({ss}) && my_node_ID in {ss.schedule} &&
! {ss.active_bit} && STATUS(! cur_bank) == INVALID]
r_ss = cur_time ; master_node_ID = {ss.master_node_ID}
psc_sf_start_next = r_ss – T_ss – psc_offset + sf_length(cur_bank)
get_sched_timer = T_GET_SCHED
MEM(! cur_bank) = {ss.schedule(NODE_ID)}
STATUS(! cur_bank, VALID) ; status_flags |= CNTDWN
switch_count = {ss.switch_count}
sf_length(! cur_bank) = {ss.sf_length}
num_schedchunks(! cur_bank) = {ss.num_schedchunks(NODE_ID)}
T_ss(! cur_bank) = {ss.T_ss}
[RCV({ss}) && my_node_ID in {ss.schedule} &&
(({ss.active_bit}) || (! {ss.active_bit} && STATUS(! cur_bank) == VALID))]
r_ss = cur_time ; master_node_ID = {ss.master_node_ID}
psc_sf_start_next = r_ss – T_ss – psc_offset + sf_length(cur_bank)
get_sched_timer = T_GET_SCHED

Fig. 2. Receiver hardware state machine for routine mode: `>routine<`

If, on the other hand, the node's MAC address (my_node_ID) *is* in the schedule, then >routine< next checks whether the "active bit" field within the SYNCSCHED frame, called {ss.active_bit}, is set. As long as the active bit is set, the node will continue to operate according to the current schedule located in the current memory bank, cur_bank. However, if the active bit is not set, then the schedule being disseminated in the SYNCSCHED frame is a newly calculated schedule that will go into effect after switch_count more superframes. That is, switch_count represents the number of remaining superframes following the current one in which the old schedule will still be used. The value of switch_count is obtained from the SYNCSCHED frame.

When >routine< encounters a SYNCSCHED frame without the active bit set, it checks the status of the reserve memory bank, !cur_bank. If the status is INVALID, then all the new synchronization and scheduling information for the new schedule has yet to be copied into the reserve memory bank, !cur_bank. After copying this information, >routine< sets this bank's status to VALID. In this way, >routine< doesn't waste effort recopying the new schedule's information into the reserve memory bank several times. That is, if >routine< encounters a SYNCSCHED frame without the active bit set but finds the status of the reserve memory bank to be already VALID, then it recognizes that it has already copied the new information into the reserve memory bank.

Routine mode ends whenever one of several error conditions occurs. For example, if a SYNCSCHED frame isn't received within the allowed time interval, then it is possible the master node has failed; thus >routine< generates the "NO_SCHED" signal and returns to the IDLE state.

4 Scheduling

4.1 The Helios Greedy Scheduling Algorithm

The master node receives an OCC frame containing packet queue occupancies from each node once per superframe. The master node may also receive a JOIN-OCC frame containing packet queue occupancies from a new node joining the network. From this information, the master node builds the $N \times C$ traffic matrix, where N is the number of nodes in the network, C is the number of wavelengths, and entry a_{ij} is the number of slots requested by node i for transmission on λ_j.

Table 3. Example traffic matrix

	λ_1	λ_2	λ_3	sum
n_1	4	1	3	8
n_2	2	3	2	7
n_3	3	2	1	6
n_4	2	3	1	6
n_5	1	1	2	4
sum	12	10	9	

Table 3 shows a traffic matrix for a network of $C = 3$ wavelengths and $N = 5$ nodes.

Helios uses a one-pass greedy scheduling algorithm, the pseudocode for which is given in Alg. 1. The algorithm creates a schedule from $t = 0$ forward in time without backtracking, always attempting to schedule the highest priority node on the highest priority wavelength. Higher priority is assigned to nodes (respectively, wavelengths) that have higher corresponding row-sums (respectively, column-sums) in the traffic matrix. In the sample traffic matrix in Table 3, the nodes have been renumbered in order of largest row-sum to smallest, such that n_1 has the largest row-sum and n_N has the smallest, with ties being broken arbitrarily. The same was done for the wavelengths: λ_1 has the largest column-sum and λ_C has the smallest. The traffic matrix gives rise to two lower bounds on the schedule length. The maximum column-sum is the *channel bound*; a schedule can be no shorter than the total demand for any one wavelength. The maximum row-sum plus C tuning latencies is called the *node bound*; to meet the demand of n_1, a schedule must be at least long enough for n_1 to transmit all its traffic and tune to each of the $C = 3$ wavelengths. The maximum of the channel and node bounds is the greatest lower bound on the schedule length.

The original scheduler developed in a previous work at NCSU ([11], [12]) produces schedules very close to the lower bound in length, but requires a prohibitively long runtime. In particular, the original scheduler has a worst-case runtime of $O(CN^{4)}$. The scheduler developed for Helios is a straightforward greedy scheduler that has a worst-case runtime of $O(C^2N^2)$. This speedup is substantial because the number of nodes is expected to be much larger than the number of channels. Moreover, the greedy scheduler can be readily implemented in hardware, resulting in an additional gain in speed. To achieve these gains in speed and simplicity, the new scheduler produces schedules that are not as close to optimal as those produced by the original scheduler. However, the greedy scheduler's results are "reasonably close" to optimal: in simulations with various patterns of network traffic demand, the greedy scheduler produces schedules within 5% of the lower bound, approximately 95% of the time.

The histogram shown in Fig. 3 corresponds to a network of 50 nodes, in which each node determines its demand for each wavelength by drawing from the same distribution (here, equally likely over the set {0,1,...,20}). For each set of traffic demands, we examined the ratio of the length of the schedule generated by the greedy scheduler to the lower bound. The histogram was created from 100,000 replications. The height of each box shows the number of replications in which the ratio fell within the range indicated. For example, nearly 58,000 or 58% of the replications resulted in ratios between 1.00 and 1.01. Furthermore, in 95% of the replications, the new scheduler produced a schedule that was no more than 3% longer than the lower bound (corresponding to ratios between 1.00 and 1.03).

Algorithm 1 The helios greedy scheduler

```
{* initialize each entry in the schedule to 0 *}
for $t = 0$ to $2glb$ do {* schedule length won't exceed 2×greatest-lower-bound *}
  for $\lambda = 1$ to $C$ do
    schedule[$t$][$\lambda$] $\leftarrow 0$
  end for
end for
{* initialize remainingDemand to the sum of all the $a_{n\lambda}$'s *}
remainingDemand $\leftarrow 0$
for $\lambda = 1$ to $C$ do
  for $n = 1$ to $N$ do
    remainingDemand $\leftarrow$ remainingDemand + $a[n][\lambda]$
  end for
end for
{* begin scheduling at first slot *}
$t \leftarrow 0$
while remainingDemand $> 0$ and $t < 2glb$ do {* there is still unmet demand *}
  for $\lambda = 1$ to $C$ do
    if schedule[$t$][$\lambda$] $= 0$ then {* if no task has been assigned to this $\lambda$, this slot *}
      $n \leftarrow 1$
      while $n \leq N$ and (unavailable[$n$][$t$] $= 1$ or $a[n][\lambda] = 0$) do
        $n \leftarrow n + 1$ {* find an available node with unfulfilled demand on this $\lambda$ *}
      end while
      if $n \leq N$ then
        for $i = t$ to $t + a[n][\lambda] - 1$ do
          schedule[$i$][$\lambda$] $\leftarrow n$
        end for
        for $i = t$ to $t + a[n][\lambda] - 1 + tuneLatency$ do
          unavailable[$n$][$i$] $\leftarrow 1$
        end for
        remainingDemand $\leftarrow$ remainingDemand $-a[n][\lambda]$
        $a[n][\lambda] \leftarrow 0$
      end if
    end if
  end for
  $t \leftarrow t + 1$ {* move to next slot *}
end while
```

4.2 Multicast

The `Helios` network nodes are equipped with fast tunable transmitters and slowly tunable receivers to form what is known as a FTT-STR architecture. For functions such as packet transmission and scheduling which operate at fine time scales (i.e., on the order of packet transmission times), the lasers are considered tunable and the receivers are considered fixed-tuned. The tunability of optical receivers is invoked only at longer time scales (i.e., on the order of seconds or hundreds of milliseconds) to address the issues of load balancing and multicast. In other words, we distinguish two regions of network operation: during the

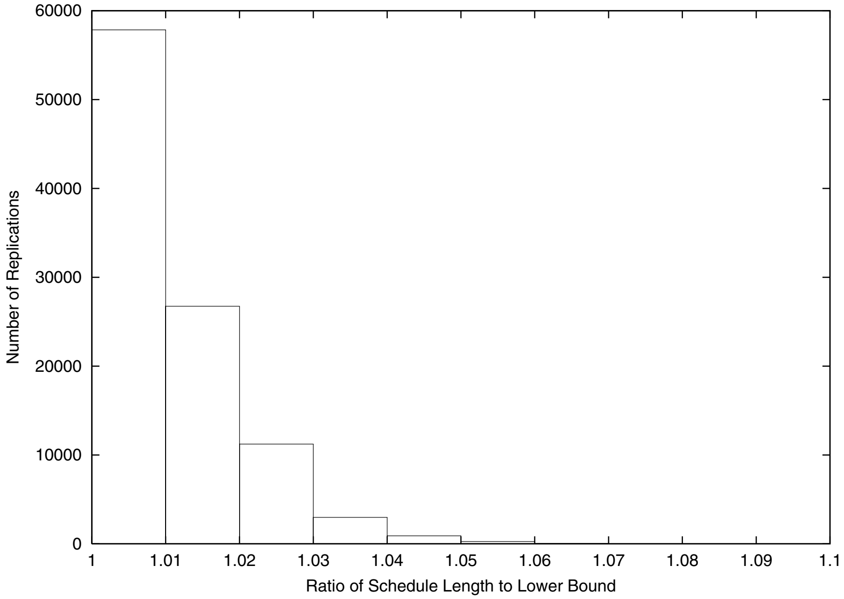


Fig. 3. Performance of the greedy scheduler in a `Helios` network of 50 nodes

normal operation phase, the optical receivers remain fixed-tuned to their home channels, while during the *reconfiguration* phase [3], the receivers are slowly retuned to new home channels in order to optimize the network for the next normal operation phase.

Let us assume that we have some information regarding the long-term multicast traffic demands in the network, including the number and composition of multicast groups, and let us further assume that this information is collected using the Helios protocol implemented at each node. Then, the problem of supporting multicast traffic in a FTT-STR broadcast WDM architecture is an optimization problem, whereby optical receivers must be assigned home channels such that a performance metric is optimized. The performance metric of interest in `Helios` is the *multicast throughput*, defined as the number of multicast completions per unit time, where a multicast completion refers to the transmission of a multicast packet to all members of its multicast group. We refer to this problem as the *multicast wavelength assignment* (MWA) problem, and we have shown in [13] that it is NP-hard.

The complexity of the MWA problem derives from two conflicting objectives that must be simultaneously satisfied. On the one hand, it is important to balance the traffic load across the different channels, while on the other hand it is desirable to assign receivers in the same multicast group to the same home channel to keep the multicast throughput high (otherwise, a multicast packet has to be transmitted multiple times, once to the home channel of the various receivers in its group). The problem is further complicated by the fact that multiple groups may not be disjoint, i.e., a given receiver may be part of multiple groups.

We have developed a number of heuristics for the MWA problem, which are described in detail in [13]. Here we provide a summary of their operation. The `Join` class of heuristics starts with each of the N receivers assigned to a separate channel, and repeatedly joins the receivers from two different channels by assigning them to a single channel, until the number of home channels is equal to the number $C, C < N$ in the network. The `GreedyJoin` heuristic applies a greedy rule in joining two sets of receivers, while the `RandomJoin` heuristic randomly joins two sets at each step. The `Split` class of heuristics starts with all N receivers in the network assigned to a single home channel, and then repeatedly selects one receiver to assign to one of the other $C-1$ channels. The `Join` class and `Split` class of heuristics take advantage of the *monotonicity* properties of the multicast throughput that were first derived in [10]. The `MLPT` heuristic takes a different approach. It first uses the LPT (Largest Processing Time) scheduling algorithm, which provides good load balancing, to come up with an initial wavelength assignment, which it then improves through an iterative approach. Based on a wide range of results in [13], the `GreedyJoin` heuristic appears to provide the best approach for the MWA problem.

4.3 DiffServ Support in the `Helios` Architecture

The basic `Helios` scheduling algorithm is appropriate for best-effort traffic but does not provide any QoS guarantees. We have modified this scheduling algorithm [4] to provide native support for the differentiated services (DiffServ) architecture currently being standardized by the IETF. Providing bandwidth and/or delay guarantees in a multiwavelength environment is an inherently complicated task, due to the need to coordinate packet transmissions among the nodes across multiple wavelengths while at the same time attempting to meet packet deadlines; the problem becomes all the more difficult when the transmitting nodes have to account for non-negligible tuning delays. We provide a brief summary of the scheduling algorithm here; details and numerical results are available in [4].

The algorithm consists of two steps. First, an initial schedule is built based on traffic reservations for the two classes of DiffServ traffic that require bandwidth and/or delay guarantees, the Expedited Forwarding (EF) class and the Assured Forwarding (AF) class. This schedule is such that all nodes can meet the QoS guarantees for their EF and AF traffic. This initial schedule is then extended to assign transmission slots for best-effort (BE) traffic, using an algorithm that ensures two important properties in the final schedule: first, that the QoS of the EF and AF traffic is not compromised for any node; and second, that best-effort transmissions are assigned to the various nodes in a *max-min* fair fashion. This latter property guarantees that the excess bandwidth in a `Helios` network is allocated fairly among the network flows. Another important feature of our guaranteed-service scheduling algorithms is that they require only small changes to the basic `Helios` scheduling algorithm. Numerical results in [4] using our

WDM simulator (see below) indicate that the algorithm works as expected and can provide QoS guarantees compatible with the DiffServ framework.

A significant contribution of our work was the implementation of a highly extensible simulator for evaluating the performance of the scheduling algorithms. Our simulator builds upon the functionality provided by the DiffServ model contributed by Nortel Networks to the popular simulator tool `ns-2`. Before our work, `ns-2` lacked support for WDM (i.e., multi-channel) links. Our WDM simulator was integrated into `ns-2` by mapping a model of a `Helios` node into an `ns-2` topology. The details of the mapping can be found in [4], while the computer code is available at [2] and can be easily incorporated into an existing `ns-2` installation. We believe that our simulator addresses an important need and we hope that it will be useful to other researchers in the field.

5 Conclusion

In this article we have presented a WDM all-optical broadcast architecture for a local area network with an accompanying signaling protocol and control algorithms. We've demonstrated how elements of DiffServ (QoS) and multicast can be easily incorporated into the architecture, both essential features for local area networks of the future.

We believe the `Helios` architecture to be a viable concept for all-optical networks of the future. Features such as fault-tolerance, the ability to support more nodes than wavelengths, and scheduled gated access to the medium combine to make this architecture a flexible framework into which, by replacing only the scheduler, new features can easily be incorporated. Our work on `Helios` continues. We plan to implement an emulation of the protocol running on commodity hardware to test various approaches to scheduling and signaling, in order to validate the concept even further.

References

1. The NGI Helios project. In *http://helios.anr.mcnc.org/*.
2. WDM support in ns-2. In *http://www.csc.ncsu.edu/faculty/GRouskas/NS/*.
3. Ilia Baldine and George N. Rouskas. Traffic adaptive WDM networks: A study of reconfiguration issues. *IEEE/OSA Journal of Lightwave Technology*, 19(4):433–455, April 2001.
4. Sudhin Bengeri. Differentiated services support for the Helios optical WDM testbed. Master's thesis, North Carolina State University, http://www.lib.ncsu.edu/etd/public/etd-16201418610131981/etd.pdf, August 2001.
5. E. Hall et al. The Rainbow-II gigabit optical network. *IEEE Journal Selected Areas in Communications*, 14(5):814–823, June 1996.
6. M. Kuznetsov et al. A next-generation optical regional access network. *IEEE Communications*, 38(1):66–72, January 2000.
7. R. E. Wagner et al. MONET: Multiwavelength optical networking. *Journal of Lightwave Technology*, 14(6):1349–1355, June 1996.

8. O. Gerstel, B. Li, A. McGuire, G. N. Rouskas, K. Sivalingam, and Z. Zhang (Eds.). Special issue on protocols and architectures for next generation optical WDM networks. *IEEE Journal Selected Areas in Communications*, 18(10), October 2000.
9. B. Mukherjee. WDM-Based local lightwave networks Part I: Single-hop systems. *IEEE Network*, pages 12–27, May 1992.
10. Zeydy Ortiz, George N. Rouskas, and Harry G. Perros. Scheduling of multicast traffic in tunable-receiver WDM networks with non-negligible tuning latencies. In *Proceedings of SIGCOMM*, pages 301–310, September 1997.
11. George N. Rouskas and Vijay Sivaraman. Packet scheduling in broadcast WDM networks with arbitrary transceiver tuning latencies. *IEEE/ACM Transactions on Networking*, 5(3):359–370, June 1997.
12. Vijay Sivaraman and George N. Rouskas. A reservation protocol for broadcast WDM networks and stability analysis. *Computer Networks*, 32(2):211–277, February 2000.
13. Dhaval Thaker. Multicasting in a partially tunable broadcast WDM network. Master's thesis, North Carolina State University, http://www.lib.ncsu.edu/etd/public/etd-120143410141221/etd.pdf, May 2001.

Service and Network Management Interworking in Future Wireless Systems

V. Tountopoulos, V. Stavroulaki, P. Demestichas, N. Mitrou, and M. Theologou

National Technical University of Athens
Department of Electrical Engineering and Computer Science
Division of Communication, Electronic and Information Engineering, Telecommunications Laboratory
9 Heroon Polytechneiou Street, Zographou 15773, Athens, GREECE
Tel: + 30 1 772 14 95, Fax: + 30 1 772 25 34
vttounto@telecom.ntua.gr

Abstract. The need for seamless communications in future wireless networks imposes the development of distributed management systems, in which we can distinguish between different business entities, the Service and Network Providers. Such a highly dynamic environment enables the Service Providers to associate with multiple Network Providers and choose the best ones, according to user satisfaction and network's reward criteria. In this direction, this paper presents a *Service Management System* from the SP's perspective *(SP-SMS)*. The SP-SMS can be separated into two planes, namely the service and network resource plane, and three layers, the *Session Configuration Layer*, the *Local Planning Layer* and the *Global Planning Layer*. Our study focuses on the Local Planning Layer and the role of the related components, which are introduced to describe the functionality of this layer. Through representative results, we demonstrate the superiority of the proposed model, in terms of utilisation and aggregate revenue.

1 Introduction

The liberalisation of telecommunications market and the success of future wireless systems require the provision of sophisticated and demanding services. The attainment of seamless communication imposes the adaptability of service requirements to the characteristics of the involved networks, without violating the limitations of physical infrastructure. The use of such services and applications usually requires a minimum Quality of Service (QoS), but the adoption of flexible resource allocation mechanisms comes up as an extremely challenging and interesting perspective.

It is obvious that a distributed, decentralised architecture for future wireless cellular systems is absolutely necessary to come up with stringent application requirements and dynamically varying available resources. The distinction between different entities in such systems enables the coupling of difficult radio infrastructure with the adaptability and scalability of multimedia service characteristics. In this direction, the introduction of Service Provider (SP) and Network Provider (NP) entities contributes to an efficient and aspiring approach to the problem.

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 899-910, 2002.

The scarcity of the radio resources and the versatility of the environment conditions, i.e. the time-variant traffic load, mobility levels and interference conditions, lead to degradations on the quality levels of the offered services. Thus, the competition between various SPs and NPs enables the effective exploitation of the cellular infrastructure, as well as the satisfaction of users' demands and requirements.

The scope of this paper is to present parts of the functional architecture of a *Service Management System* from the SP's perspective (*SP-SMS)*, focusing on the functionality of the *Local Planning Layer.* Such a system enables the dynamic co-operation of an SP with the most appropriate NPs, in terms of user satisfaction and cost efficiency, in a competitive environment. More specifically, it addresses the problem of *Service Configuration and Network Provider Selection* (*SCNPS*), which will be introduced as part of the local planning layer components' logic.

The rest of the paper is organised as follows: Section 2 presents the general assumptions for a future wireless management system, regarding the policies of SPs and NPs. Section 3 focuses on the functionality at the local planning layer, giving a mathematical approach to the SCNPS problem. Section 4 presents a set of results for the evaluation of the proposed architecture. Finally, Section 5 includes concluding remarks.

2 System Model

This section presents briefly the objectives of the different entities (SPs and NPs) in a future wireless cellular system. It, also, gives the general structure of the proposed management system and describes the functionality of the involved layers.

2.1 SP Objectives and Policies

The SP role is to provide users with a set of services. Each SP has a volume of users, who have been subscribed to some or all the services available in this SP and have been sorted out into classes (user classes), according to their preferences in priority access to services. Each subscriber has been associated with a service usage profile, which models the behaviour exhibited by a typical subscriber of the user class, with respect to a specific service.

The users of each class have a set of permissible quality levels associated with each service, offered from the SP. Each quality level has been assigned a maximum tolerable price (tariff) and a utility measure. This factor is used to express the preference of a user class to some quality levels with respect to other permissible ones. It is assumed that the information, regarding permissible quality levels, maximum prices, utility levels, etc., is stored in the profiles of the different user classes.

Of the main objectives of an SP is to determine a reference quality level for a given service, which represents the minimum bandwidth required for the support of this service. During a session, the SP can allocate the service to a higher quality level, if this is acceptable. On the other hand, the SP is responsible for the accommodation of users to the most appropriate NP, according to quality and cost criteria, which can secure at least the reference quality level for the support of the selected service.

2.2 NP Policies

The accomplishment of the SP role requires the co-operation with NPs. The NP role is to offer the network-level connectivity necessary for supporting the services. Within a portion of the service area, each NP can support up to a certain number of SP subscribers per quality level and imposes a tariff for the provision of a service at a quality level. The tariffs are determined by the NP policies, taking into account parameters, such as the managed network status, the volume of resources that the SP uses (either globally or in a specific area of the network), the area of the network, and the time zone.

2.3 SP-SMS Design

Without losing generality, we examine the SP-SMS management system from the perspective of a single SP. Figure 1 shows the general framework of a future wireless management system, depicting clearly the distinction between the different planes and layers, described below.

A future wireless management system can be classified into two planes, the *service plane* and the *network resource plane*. Both SPs and NPs are involved in the *service plane*. Through the functionality that is framed therein, the SP is enabled to find, in each service area portion and time zone, the target quality levels and the best NPs, for each service and user class. The SP-SMS covers the SP-related parts of the service plane.

The network resource plane involves only NPs. At this plane, the NPs manage the resources of the network infrastructure so as to meet the agreements having been established with SPs. This plane can rely on legacy network management systems. In this respect, the service plane functionality of the NPs can be seen as a necessary extension to legacy network management systems. The extension will enable the interworking with SPs and the promotion of the NP infrastructure.

As it can be seen from figure 1, the SP-SMS is structured into three layers, namely the *Session Configuration Layer,* the lower one at the SP-SMS hierarchy, the *Local Planning Layer* and the *Global Planning Layer*. Each layer contains a component type, which can be differentiated from the other component types of the other layers by the time scale and the service area portion, on which it operates. It should be noted, here, that the components of the SP-SMS co-operate with corresponding entities of the NP that fall within the service plane. In the rest of this subsection, we describe the functionality of the different components.

The *SP Session Configuration Components* (*SP-SCC*s) are enabled to handle with the incoming sessions. They are responsible for the monitoring of the network performance and the notification for the modification of the network configuration, when this is needed. Each SP-SCC controls a subset of the service area and is controlled by the components of the higher layers. The SP-SCC configuration specifies for each service and user class the target quality level that should be offered by the NPs, the best ones that can offer the target quality level and the maximum demand volume that each of them can accommodate.

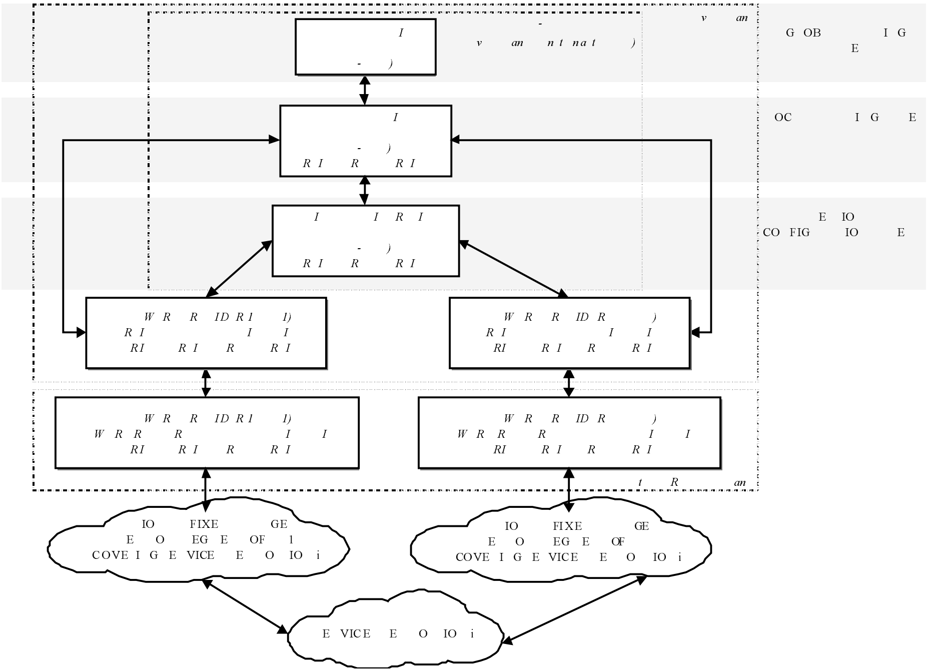


Fig. 1. Layers, components and high-level distribution pattern in the management framework

The *SP Local Planning Components* (*SP-LPC*s) control the counterpart portions of the service area as the SP-SCCs. They are responsible for the reconfiguration of the SP-SCCs, as it is described in section 3.

Finally, the *SP Global Planning Components* (*SP-GPC*s) focus over large subsets or the entire service area, and assist the components in the underlying layers in accomplishing their roles, by providing global policies and information.

3 Local Planning Layer

This section describes the functionality of the Local Planning Layer of the SP-SMS and the role of its components. More specifically, it addresses the problem of Service Configuration and Network Provider Selection (SCNPS), in order to clarify the concept of this layer in a future wireless management system. First, we present the possible structure of the management system, introducing the input parameters for our model, and then we give a mathematical presentation of the problem.

3.1 High-Level Description of the SP-LPC Functionality (SCNPS Problem)

Without loss of generality, we focus on a portion of the service area, denoted as i. As it has been stated above, there is an SP to control this area, which can be associated with a set of NPs, denoted as NP_{avail}. In service area i, we assume that the SP can offer a set S of services to a set UC_s of user classes, which have been subscribed to the service $s \in S$. We, also, define a set $Q_{s,uc}$ of permissible quality levels for the service s, which can be accessed by the subscribers of class $uc \in UC_s$. In the user's profile, it has been defined that the maximum acceptable price and the corresponding utility measure, when the service s is offered to the users of class uc at quality level q ($q \in Q_{s,uc}$), are $\overline{c}_{s,uc,q}$ and $u_{s,uc,q}$, respectively.

We assume that each $np \in \mathrm{NP}_{\mathrm{avail}}$ can accommodate the specifications of the set of services S. There are two aspects that can configure the policy of an np. First, the maximum network traffic load volume, $L_{\max}(np)$, that the network provider can support in the service area portion i. Second, the set of tariffs, $c(np, s, q)$, that the np will impose for the support of service s at quality level q ($q \in \bigcup_{uc \in UC_s} Q_{s,uc}$). These tariffs can depend on the network status, the resources used by the SP, the area of the network and the time zone.

Users that are found in service area portion i send session requests to the controlling SP-SCC (SP-SCC-i). In response to the session request, the SP-SCC-i indicates the target quality level q and the best np for each service. The SP-SCC-i reply is straightforward, based on the user information and on its configuration that has been done by the SP-LPC-i. The SP-SCC-i monitors whether its configuration is appropriate for the service demand. In this respect, the SP-SCC-i monitors the demand volume per service s and user class uc, as well as the actual quality levels $Q_{s,uc}$ offered by the NP_{avail}, in the area i. In case the SP-SCC-i identifies that its configuration is not appropriate for handling the demand for one or more services of the set S, it invokes the SP-LPC-i and requests a modification. In response to the SP-SCC-i invocation, the SP-LPC-i must apply the SCNPS problem for modifying the SP-SCC-i configuration.

The service demand pattern can be modelled through a vector $D_s = \{d(s,uc) | \forall (s,uc) \in (S \times UC_s)\}$. Each element of the vector, $d(s,uc)$, corresponds to the demand for service s that originates from the users of class uc and should be accommodated by the SP, within the service area portion i and time zone.

3.2 Mathematical Formulation of the SCNPS Problem

The SCNPS problem has two general objectives. First, to compute an allocation of the service demand pattern to quality levels, $A_{QL} = \{ql(s,uc) | \forall (s,uc) \in$

$(S \times UC_s)\}$. Each element of the allocation, $ql(s,uc)$ ($ql(s,uc) \in Q_{s,uc}$), is the target quality level, at which the users of class uc should access service s for the specific time zone and service area portion. The second objective of the SCNPS problem is to compute an allocation of the service demand pattern to network providers, $A_{NP} = \{r(np,s,uc) | \forall (np,s,uc) \in (NP_{avail} \times S \times UC_s)\}$. Each element of the allocation, $r(np,s,uc)$, denotes the part of the demand for service s, corresponding to the users of class uc, that should be satisfied by network provider np ($np \in NP_{avail}$). Network provider np should satisfy the service demand portion at the selected quality level $ql(s,uc)$. It holds that $r(np,s,uc) \leq d(s,uc)$.

The objective function that should be optimised by the allocations is denoted as $OF(A_{QL}, A_{NP})$, and is associated with the utility measures and the costs achieved by the allocations. Our target is to maximise this objective function, which can be expressed as:

$$OF(A_{QL}, A_{NP}) = \tag{1}$$

$$\sum_{s \in S} \sum_{uc \in UC_s} d(s,uc) \cdot u_{s,uc,ql(s,uc)} - \sum_{s \in S} \sum_{uc \in UC_s} \sum_{np \in NP_{avail}} r(s,uc,np) \cdot c(np,s,ql(s,uc))$$

The allocations (selected quality levels and network providers) should respect some sets of constraints. The first one imposes the satisfaction of the service demand pattern and it can be expressed as:

$$\sum_{np \in NP_{avail}} r(np,s,uc) = d(s,uc), \forall (s,uc) \in (S \times UC_s) \tag{2}$$

The second set of constraints guarantees that the capacity limitations of the selected NPs will not be violated. In this point, we introduce the function $L(np, A_{QL}, A_{NP})$, which represents the load that will be imposed on network provider np ($np \in NP_{avail}$) as a result of the allocations A_{QL} and A_{NP}. So:

$$L(np, A_{QL}, A_{NP}) \leq L_{\max}(np) \; \forall np \in NP_{avail} \tag{3}$$

Another set of constraints should guarantee that the tariffs that will be imposed to the users should be compliant with the specifications in their profiles. This can be expressed as

$$c(np,s,ql(s,uc)) < \bar{c}_{s,uc,q_s(uc)}, \forall (s,uc) \in (S \times UC_s), \tag{4}$$

$$\forall np \in NP_{avail} : (r(np,s,uc) > 0)$$

Here, it should be noted that the solution of allocation of the service demand to quality levels and network providers will be forwarded to the SP-SCC-i, SP-GPC and the service plane mechanisms of the chosen NPs. The SP-SCC-i will use this new solution for continuing the work at the session configuration layer. The SP-GPC will update its global view of the manner, in which users are served and services are provided in the service area. The whole problem, as it was described in this section, may have the graphical presentation of figure 2.

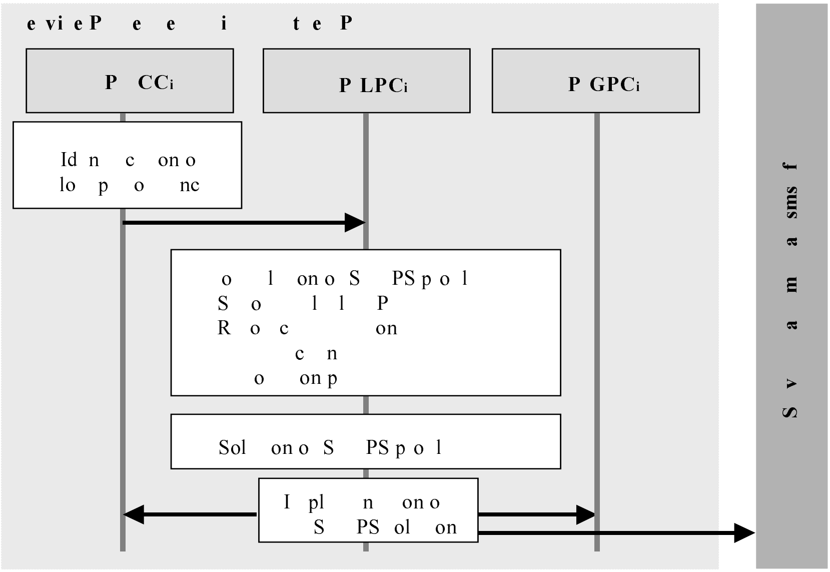


Fig. 2. Description of the Local Planning Layer functionality

4 Results

In this section, we intend to present a set of results for the evaluation of the proposed management techniques. More specifically, we analyse the problem of NP selection for the support of services in a future wireless cellular system, addressing two major possible solutions, the degradation in the quality of offered services and the binding of resources from alternative network providers. These solutions are evaluated according to the objective function, as it was described in the previous section. This function simulates the behaviour of the SP, according to the policies of the available NPs, as well as it takes into consideration the users' preferences, as they have been stored in their profiles. Finally, we evaluate the results using the concept of channel utilisation, which can be defined as CU = (1-P)*A/n, where P is the blocking probability, A is the offered load and n is the number of channels.

For the purposes of this paper, we assume that the SP can choose between two NPs, np_1 and np_2 respectively, in order to engage resources for the support of a specific kind of service, which is not appropriately served within a portion of the service area. This service can be related to two quality levels, denoted as QL_1 and QL_2 respectively, both of them corresponding to 1% blocking probability. During the subscription of a user to a certain user class of the SP, the SP becomes aware of the user's profile and it can, therefore, estimate a utility measure for the allocation of the user to a specific quality level. Table 1 summarizes some typical values for the input parameters of the SCNPS problem.

Table 1. SP and NP policies for the SCNPS problem

	Utility Measure		**Cost /Tariff**		**Bit Rate (Kbps)**
	UC_1	UC_2	NP_1	NP_2	
QL_1	5	-	1.5	3.5	384
QL_2	5	2.5	0.75	1.75	144

It should be noted here that we take into consideration the structure of a CDMA cellular system, with typical values of the channel bandwidth $\Delta f = 5MHz$ and chip rate $CR = 4.096Mcps$. We, also, make the assumption that the SP has initially bound 16 channels from the cheapest NP. All users, who have already entered the system, have been accommodated to np_1, which appears to be cheaper than np_2, and they have been allocated at the high quality level QL_1. In an instance of time, the offered load in the area, which is approximately $OfferedLoad = 12Erlang$ and can be shared equally to both of classes, that is $Tr_{UC_1} = 6Erlang$ and $Tr_{UC_2} = 6Erlang$, performs a 5% blocking probability, which means that the $SP - SCC_i$ has to request for a modification in configuration. There are two possible solutions:

- np_1 has no more resources to allocate, so the SP determines to degrade the quality in low priority users. In this case, the new traffic in the area will be modified to $OfferedLoad_{NEW} = 8.5Erlang$. Then, the objective function value is: $OF = 31.5$.
- The SP can buy resources from np_2. Our target is to keep the blocking probability at $B = 1\%$. For this reason, we need to transfer an amount of traffic from the low priority class UC2 to np_2, in order to keep all of them in the high quality level. From Erlang B table, we can see that for 16 channels and $B = 1\%$ the offered load in np_1 must be approximately 9 Erlang, from which an amount of 6 Erlang refers to the traffic of UC_1 and 3 Erlang to the traffic of UC_2. The rest 3 Erlang

of the traffic of UC_2 will be served from np_2. Thus, the objective function will become: $OF = 36$.

We now examine the effect of the tariff policy of np_2 in the objective function. On figure 3, we have declared 5 schemes of different tariff policies for np_2 and the respective objective function for the case that the SP buys resources from that NP. On the same figure, we have depicted the objective function for the case of degrading the quality in low priority users, which remains as is for all schemes, since the alterations in the tariff policy of np_2 have no effect on that case.

As it is illustrated from this figure, the solution of buying resources from an alternative NP (SP-SMS approach), keeping all the connections in the high quality level, is superior than the other one, that implies the degradation of the low priority connections, in order to keep all of them in the same NP. Only if the difference between the cost policies of the various available NPs exceeds a threshold, the proposed solution is not profitable. This threshold implies that the case of degradation the low priority users is preferable from SP's perspective, when np_2 becomes at about 350% more expensive than np_1. However, it would be safe to say that this threshold seems difficult to be reached, unless in very scarce circumstances.

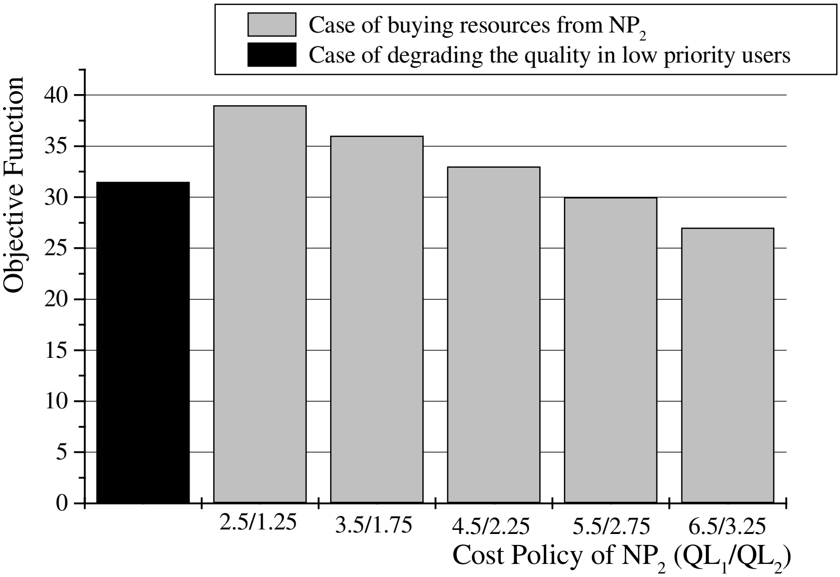


Fig. 3. The objective function vs. the cost policy of np_2

We, now, examine the effect of utility measure on the SCNPS problem. Specifically, we assume that the SP's revenue by allocating UC_1 to QL_1 and QL_2 is

5 and 0 units respectively. We also assume that the cost values for np_1 are $c(np_1, QL_1) =$ 1.5 and $c(np_1, QL_2) =$ 0.75 and for np_2 are $c(np_2, QL_1) =$ 3 and $c(np_2, QL_2) =$ 1.5. This means that we have assumed a reasonable scenario for the tariff policies of the available NPs, which implies a double cost policy of np_2 regarding np_1. Figure 4 depicts the objective function for 4 scenarios of the utility measure and for the allocation of UC_2 to respective quality levels. From this figure, we can conclude that, when the utility measure of user classes for quality levels is high, the SP-SMS leads to better performance. As the utility measure decreases, the SP-SMS logic gives worse results than the degradation case, when the user has no strong interest for the quality levels.

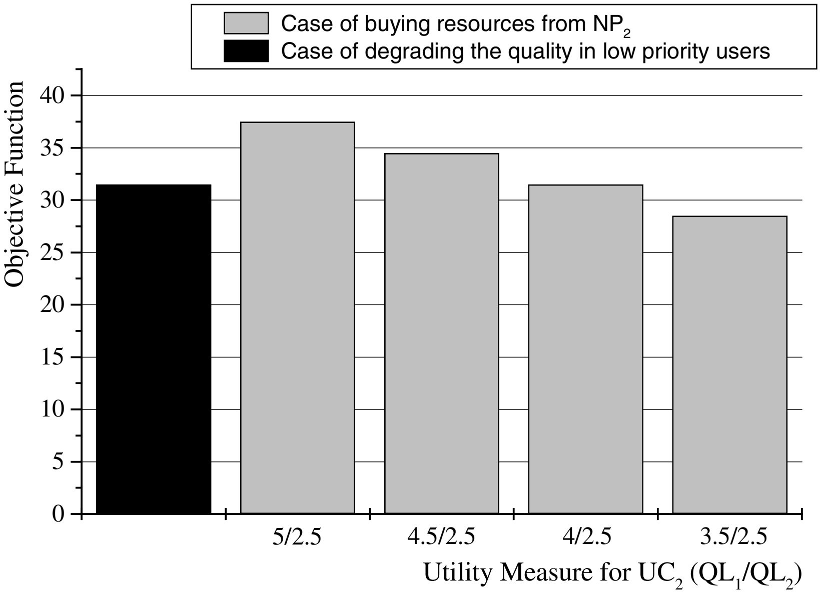


Fig. 4. The objective function vs. the utility measure of UC_2

From NPs' perspective, we examine the impact of the above scenario in the channel utilisation. The results can be found on table 2. From it, we can conclude that the capability of choice between different NPs for the support of a set of services in a specific area is much more preferable than the degradation in the quality of the offered services that some users experience. The channel utilisation is better for each NP separately, as well as the system as a whole.

Table 2. Channel Utilisation for the cases under study

	Channel Utilisation	
	NP_1	*NP_2*
Case 1	0.495	0
Case 2	0.549	0.330

5 Conclusions

The main feature of the future wireless cellular systems is the intelligence and flexibility in the provision of services. In this respect, we presented in this paper a Service Management System, from the SP's perspective, which has been classified into three layers and enables the SP to co-operate with a set of available NPs, in order to find the most appropriate one, according to cost-effective and user-satisfaction criteria.

The SP-SMS deals with the incoming sessions and allocates users' requests to different quality levels, according to the availability of resources. In case of low performance within a service area portion, there is an invocation for modification in its configuration. The objective of the management system is to find the best allocation of the service demands to quality levels and available NPs. The solution of the SCNPS problem must be communicated to the rest of SP-SMS layers and the service plane mechanisms of the associated NP.

References

1. S. Trigila, A. Mullery, M. Campolargo, J. Hunt, "Service architectures and service creation for integrated broadband communications", Computer Communications, Vol. 18, No. 11, 1995
2. S. Trigila, K. Raatikainen, B. Wind, P. Reynolds, "Mobility in long-term service architectures and distributed platforms", IEEE Personal Commun., Vol. 5 No. 4, Aug. 1998
3. "Management models for telecommunications", Feature topic in the IEEE Commun. Mag., March 1996
4. V.Garg, D.Ness-Cohn, T.Powers, L.Schenkel, "Direction for element managers and network managers", IEEE Commun. Mag., October 1998
5. IST project MONASIDRE (Management of networks and services in a diversified radio environment) Web site, www.monasidre.com, Feb. 2001
6. IST project SHUFFLE (An agent based approach to controlling resources in UMTS) Web site, www.ist-shuffle.org, Jan. 2001
7. J.-T.Park, J.-W.Baek, J.W.-K.Hong, "Management of service level agreements for multimedia Internet service using a utility model", IEEE Commun. Mag., Vol. 39, No. 5, May 2001
8. U. Varshney, "Recent advances in wireless networking", IEEE Computer, Vol. 33, No. 6, June 2000
9. A.K.Talukdar, B.R.Badrinath, A.Acharya, "Rate adaptation schemes in networks with mobile hosts", *Proc.* ACM/IEEE MobiCom'98, pp. 169-180, Oct.1998

10. Sunghyun Choi and Kang G. Shin, "Location/Mobility-Dependent Bandwidth Adaptation in QoS-Sensitive Cellular Networks," *Proc. IEEE Vehicular Technology Conference*, October 2001
11. “Design of broadband multiservice networks”, Feature topic in the *IEEE Commun. Mag.*, Vo. 36, No. 5, May 1998

Scheduling Differentiated Traffic in Multicarrier Unlicensed Systems

Giannis F. Marias and Lazaros Merakos

Department of Informatics, University of Athens, TK15784, Tel: +30107257560
{marias@mm.di.uoa.gr, merakos@di.uoa.gr}

Abstract. Over the last few years, a number of mechanisms have been proposed for scheduling different type of traffic over base stations-oriented wireless and mobile systems. The majority of these mechanisms focus on access control in the base stations-to-mobile units part of the wireless and mobile system. Recent proposals for the unlicensed spectrum in the 5GHz band redefine the problem, since base stations operated by different operators and organizations in overlapping geographical areas need access resolution mechanisms to allocate wireless resources. This issue is addressed here, and a multicarrier access control scheme, called QoS based Dynamic Channel Reservation (QDCR), is proposed. QDCR allows base stations to select the least congested available carrier, to compete for carrier reservation based on QoS requirements, and to share the allocated carrier and time with its associated Mobile Terminals (MTs).

1. Introduction

The current Internet provides only a best effort service, and it is sufficient for traditional Internet applications like web browsing and e-mail. On the other hand, several target applications require better performance than the best-effort Internet. The ATM, the Integrated Services (IntServ) [1] and more recently, the Differentiated Services (DiffServ) architecture [2], although different, can offer services over Internet (or Intranets) that go beyond the best effort. To meet the increasing demand for wireless and mobile multimedia services, future wireless and mobile networks should adopt new technologies and mechanisms in order to provide the high capacity and the QoS required to support broadband services under limited radio spectrum. During the last years, wireless and mobile ATM, and IP capable systems are nominated as the best option to extend wireline Internet or Intranet, in contrast to the traditional circuit switched voice-based solutions.

In the wireless communication area, the FCC has opened a 300 MHz for Unlicensed National Information Infrastructure (U-NII) band within the 5 GHz range. The FCC stated that in making available this spectrum it anticipates that these U-NII devices, which do not require licensing, will support the creation of new wireless services. The U-NII spectrum is allocated at 5.15-5.25GHz, 5.25-5.35GHz and 5.725-5.825GHz frequency bands, with different transmission power regulations. Devices operating with low power, such as phones and handsets will use the low U-NII band, portable devices on a SOHO environment will use the middle U-NII band, and devices operating over larger coverage areas, such as mobile phones and terminals, will use

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 911-923, 2002.

the high U-NII band. On the other hand, in Europe, the 5GHz band has been reserved by ETSI for the HIPERLAN (High Performance Radio LAN), which is under development within the Broadband Radio Access Network (BRAN) project. Depending on national regulations, 100-150MHz is reserved, but there are proposals to extend this range up to 445MHz. The HIPERLAN/2 network uses a Convergence Layer, to become fixed network independent; it will be able to provide the Quality of Service (QoS), which users expect from wired IP and ATM networks, with data transfer rates up to 25Mbps [3]. In Japan, the 5.15-5.25GHz band has been allocated by MPT to high-speed wireless access. From the FCC, ETSI and MPT regulations, it is expected that the license exempt 5 GHz band will accommodate high speed wireless ATM (wATM) and wireless IP (wIP) applications world wide.

Due to the unlicensed characteristic of the 5 GHz band, dynamic carrier allocation mechanisms are essential. These mechanisms should avoid the use of interfered carriers, especially in outdoor environments, where several operators might share the available spectrum in overlapping coverage areas. Fixed allocation method requires prior arrangements between operators, whilst dynamic allocation (e.g., DCA) does not. Moreover, DCA is more suitable than fixed allocation (e.g., FCA) methods for supporting guaranteed QoS wireless applications [4].

This paper presents a QoS based, Dynamic Channel Reservation (QDCR) architecture, suitable for Base Station (BS) oriented wATM or wIP systems. QDCR provides a set of rules followed by BSs, to regulate competition, and facilitate wireless resource reservation, based on differentiation of the expected level of service. According to QDCR, each BS, through contention periods, dynamically discovers its adjacent or co-channel interference neighbors, and competes with them taking into account differential QoS demands, traffic demands, and perceived QoS. The proposed approach requires no RF survey, and frequency pre-planning phases. Thus, providers do not have to produce interference matrices (e.g., the compatibility matrix in [5]). Moreover, the proposed method is QoS adaptive, since it takes into account the QoS of the established connections and the perceived delays for carrier reservation.

The structure of the paper is as follows. Section II discusses the assumptions that cover the wireless system, in order to apply the QDCR method. In Section III, we present the distributed contention and reservation method, which enables contention, based on service differentiation. In Section IV we describe the simulation environment and illustrate the performance of QDCR, based on the simulation results. Finally, in Section V, we summarize the conclusions.

2. System Assumptions

QDCR applies to systems where the BSs act as communication hCF, offering wIP or WATM access to MTs. Each MT maintains an association with one of the BSs, until it performs a handover. MTs do not communicate directly with other MTs (i.e., no ad-hoc features are assumed). Each BS uses a Medium Access Control (MAC) mechanism and controls the radio access of its associated MTs. The MAC could be based on any dynamic, slotted, Time Division Multiple Access/Time Division Duplex (TDMA/TDD) approach. This MAC includes a time-scheduling mechanism, which schedules the transmission of MPDUs (for both uplink and downlink directions) based on QoS requirements per connection and MT, and produces variable length time frames. Each time frame starts with a Frame Header (FH), which includes a slot map. Each associated MT reads this FH in order to determine which uplink or downlink slots in the frame are scheduled for it. The structure of the wireless MAC frame might be similar to [8], even if [8] was introduced for wATM.

We assume that the available unlicensed spectrum B (from X_0 MHz to X_1 MHz) is divided into M broadband carriers, each of W MHz. The central carrier, Fc, are given by the equation Fc=F_0-cW MHz, where F_0=X_1 MHz and c=0,1,…,M. The spectrum between Fc=F_0-W/2 MHz and Fc=F_0+W/2 MHz is the RF carrier Fc. For 1bit/hz modulation efficiency and 25 MHz channelization scheme, a throughput of 25 Mbps per carrier is provided. BSs, belonging to different operators or organizations, using one carrier of the multicarrier structure (M carriers) of the unlicensed band B, and, operating in overlapping geographical areas, should first reserve time on one carrier, before starting scheduling the reserved time to their associated MTs (through the MAC protocol). Moreover, in environments where the BSs switch among carriers, it is essential to support a mechanism that informs the MTs about the next carrier that their associated BS will use. For this purpose, we introduce a Frame Trailer (FT), transmitted at the end of the time frame. QDCR assumes that all the components (competing BSs and associated MTs) are slot synchronized, and they use the same TDMA slot length. We assume that a BSs can use one RF carrier at a time (i.e., single antenna assumption). The total number of M carriers is available for all BSs. We further assume that the BSs do not communicate directly, and that all the BSs use the same transmitted power level, PBS, and all the MTs use the same transmitted power level, PMT. Normally PMT≤PBS. For instance Tx power can be 100mW (20dB) for small indoor coverage areas, or 1000mW for outdoor larger coverage areas (i.e., HIPERLAN type 2 and U-NII middle band). To sense idle carriers, a threshold of Pth dBm (e.g. -100 dBm) is adopted for the BSs. Furthermore, omnidirectional antennas are considered for the BSs. To cope with Turn Around Times (TAT), that is time required to switch from receive to transmit mode and vice versa, we assume that all MTs require one slot. When BSs communicate with MTs (i.e., during their reservations), they use one slot for TAT; otherwise, BSs TAT is considered smaller (e.g., during competition). Likewise, for the Switch Carrier Time (SCT), that is time required to switch carrier we assume that all MTs require one slot. When BSs communicate with MTs (i.e., during their reservations) they use one slot for SCT; otherwise, BSs SCT is considered smaller (e.g., during competition periods). In indoor installations, wIP or wATM LANs might be isolated, and covered by a single operator. The isolated wLANs do not interfere with another wLAN operating in the same area. In such a case, carrier allocation can be performed dynamically, in a centralized fashion, through a central scheduling entity [9], which co-ordinates the access to the shared wireless resources.

3. QDCR Mechanism

Providing QoS services in a mobile environment requires that the radio MAC supports some degree of separation between different types of services. We propose to support three service classes: loss sensitive, delay sensitive and best effort. To provide efficient carrier selection criteria, QoS based access, and low reservation delays, QDCR uses special signal bursts broadcast by BSs. Moreover, to avoid congestion, QDCR separates control and data channels. Control channels are used to resolve contentions and to broadcast carrier status information.

3.1. Burst Signals

A signal burst is energy transmitted by BSs to indicate certain conditions and to broadcast control information. Burst signals use a higher transmitting power level, BPBS, than normal bursts, i.e., BPBS > PBS. QDCR uses the following burst signals:

- A BS transmits the Priority Burst Signal (PBS) during the priority resolution phase, declaring its QoS demand.
- A BS transmits the Request Burst Signal (RBS), during the competition phase, declaring information such as perceived delay, and reservation period request.
- A BS transmits the Periodic Priority Burst Signal (PPBS) periodically.

PBS requires less than a slot for its transmission The RBS duration is variable. PPBS is broadcast periodically, requires one slot to transmit, and the period is a system parameter.

3.2. QDCR Channels

According to QDCR, once one or more BSs sense idle carrier, the QDCR superframe starts. This superframe consists of several periods (control and data channels), allowing BSs to solve the competition, to reserve the carrier, to exchange data with their associated MTs, and to broadcast control information. The QDCR superframe channels are: a) the Priority Resolution Channel, b) the Contention Resolution Channel, c) the MAC Channel, and d) the CF Channel.

Priority Resolution Channel (PR-CH) — During this period, each BS estimates its Reservation Priority (RP). In QDCR, each BS competes with interferers in order to reserve a carrier for a time period equal to its TDMA time frame. Assume that a BS$_i$ serves K$_i$ connections, among all its associated MTs, classified as:

- {C1, C2, …, Ck$_x$} real time connections
- {V1, V2, …, Vk$_y$} non real time connections

where, $k_x+k_y=K_i$. Real time (*rt*) connections impose transfer delay requirements, whilst non real time (*nrt*) connections impose loss requirements. Each *rt* connection Ci ($0<i<k_x+1$) introduces a transfer delay violation threshold, Di^{thr}. Each *nrt* connection Vi ($0<i<k_y+1$) introduces a cell loss violation threshold, Li^{thr}. The RP for a BS$_i$ is identical to the parameter defined in [10], and given by the following equation :

$$RP = \frac{1}{2}\,\frac{\overset{k_x}{\underset{i=1}{\text{Sum}}}\,\dfrac{Di}{Di^{thr}}}{k_x} + \frac{1}{2}\,\frac{\overset{k_y}{\underset{i=1}{\text{Sum}}}\,\dfrac{Li}{Li^{thr}}}{k_y}$$

where Di is the delay that the connection Ci experiences, and the Li is the cell loss that the connection Ci experiences. The PR-CH period consists of a constant number of slots (e.g., 2 slots). This period is further divided to PR-CH minislots (p-slots). Each p-slot position corresponds to a particular RP. For instance assuming a 5 p-slot granularity, the first p-slot of the PR-CH period corresponds to RP≤0.2, the second p-slot corresponds to 0.2<RP≤0.4, and the last p-slot corresponds to 0.8<RP≤1, as shown in figure 1.

According to the estimated RP_i, the BS_i will broadcast its PBS_i within the corresponding p-slot. Best effort service class will use the first p-slot of the PR-CH period to broadcast the corresponding PBS. If D_{PBS} is the duration of signal PBS, and D_{PS} is the duration of p-slot, then $D_{PS}>D_{PBS}$, and $TAT<D_{PS}-D_{PBS}$. This allows BS to switch from transmit to receive mode and sense PBS broadcast on the next order p-slot. A backlogged BS, i.e., with low PR, sense the PBS burst of the BS illustrating higher PR, because the latter will broadcast its PBS using a higher order p-slot. Backlogged BSs should select new carrier, among the M candidates, to compete for it.

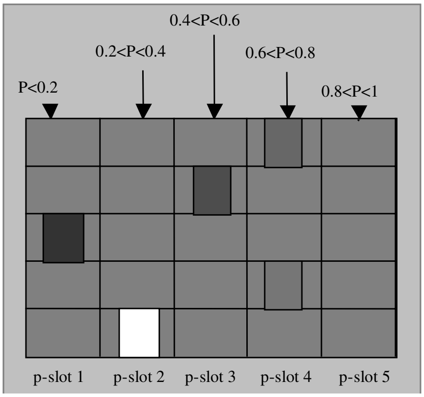


Fig. 1. The BSs broadcast their priorities through PBS bursts on the corresponding p-slot of the PR-CH. The wining BSs broadcaston the higher order p-slots

Contention Resolution Channel (CR-CH) — On the PR-CH channel we have adopted a structure (i.e., number of p-slots) to represent RPs with certain granularity. Thus, it is possible for two more BS to use p-slots of the same order to broadcast their PBSs, even if their RPs have different values (e.g., on the 2nd decimal digit of RPs). To overcome this problem we introduce the CR-CH. During CR-CH period each BS, survived from priority resolution phase, broadcast its reservation requests (through the RBS), and realizes the reservation requests of other interfering survivors. Reservation requests represent either current MAC frame time length, or mean reservation delay, or both. The CR-CH comprises of an integer, but not fixed, number of slots, each of which is divided to a fixed number of minislots (c-slots), as figure 2 illustrates. The RBS signals are transmitted on continuous c-slots, and simultaneously by the competing BSs. We introduce a granularity factor g, 0<g<1. If T slots is the reservation request (MAC length, delay, or both) in slots, then RBS will use [g*T] c-slots for its transmission. If D_{CS} is the duration of c-slot, then $TAT<D_{CS}$. This allows BS to switch from transmit to receive mode and sense RBS broadcast by another BS.

Longest Job First (LJF). According to this discipline the winner of the competition is the BS with the larger reservation request (i.e., having the larger MAC frame). Thus, if TFD_i is the number of slots the BS_i wishes to reserve on this carrier (i.e., current time frame length), then the BS_i transmits a RBS_i of $[g^*(TFD_i)]$ c-slots. The winner (survivor) BS is the one that broadcasts the larger RBS

Delayed Job First (DJF). The winner of the competition is the BS that received the highest mean delay from its previous reservation attempts on any carrier. Each BS_i records the last reserved slot in any carrier, say T_{Ri}, and switches to a carrier in order to compete for it. Then it calculates the T_M=mean(T_{Ri}). If the BS_i is involved in the competition, BS_i transmits a RBS_i signal, equal to $[g^*(T_M)]$ c-slots. The mechanism is identical for the LJF and DJF disciplines. In the former case the RBS is proportional to the frame size, whist in the latter case the RBS is proportional to the received reservation delay.

Delayed and Longest Job First (DLJF). This is a combination of LJF and DLF disciplines. The winner of the competition is the BS that experiences the larger combined reservation delay, and time frame. Thus, if T_{Ri} is the last reserved slot of a BS_i in any carrier, and the contention for a carrier involves the BS_i, and TFD_i is the number of slots the BS_i wishes to reserve on this carrier (i.e., current time frame length), then the BS_i transmits a RBS_i of $g*(TFD_i+T_M)$ c-slots, where T_M=mean(T_{Ri}). A backlogged BS, i.e., with low reservation request, sense the RBS burst of the BS with higher reservation request, because the latter will broadcast an RBS using at least on more c-slot. Backlogged BSs should select new carrier, among the M candidates, to compete for it. The survivor is the BS that has completed its RBS transmission, switched on receive mode and senses the carrier idle. This BS will reserve the carrier.

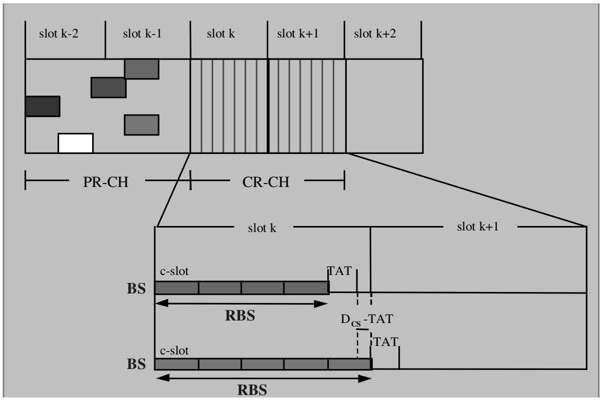


Fig. 2. The Contention Resolution (CR-CH) structure

Medium Access Control Channel (MAC-CH) — This period is used for data transfer, i.e., accommodates the MAC frame. It consist of

- *Frame Header Broadcast Channel (FHB-CH),* Within this channel the BS broadcast its MAC frame slot map to its associated MTs. Moreover, each BS, broadcasts a keyword (as a unique identifier) within the FH This key is exchanged between BSs and MTs during the association phase, and it is used by the MTs in order to identify if the winner of a competition on a carrier is their associated BS.
- *Down Link Data Channel (DLD-CH),* with variable duration, accommodating information sent through BS to MTs.
- *Turnaround Channel (T-CH),* which occupies one slot and allows MTs or BSs to switch from receive to transmit mode and vice versa. It is used more than one time per QDCR superframe.
- *Up Link Data Channel (ULD-CH),* with variable duration, accommodating information sent from MTs to other MTs or fixed terminals through the BS.
- *MTs Contention Channel (MC-CH),* which allows associated MTs, with no allocated ULD-CH slots, to request reservation slots, or accommodates registration or re-association requests from MTs performing registration or handover. The number of MC-CH slots can be static or dynamic.
- *Frame Trailer Channel (FT-CH),* which occupies one slot. Within FT-CH the BS broadcast the FT to the associated MTs. FT includes a visiting list of the carriers that the BS will visit sequentially until a successful reservation. The construction of the visiting list is based on Selection Parameters, discussed later..

Periodic Priority Resolution Channel (PPR-CH) — This channel uses one slot, and during this period the BSs broadcast their priorities. This channel is similar to PR-CH of the competition period. The difference is that only the sensing BSs broadcast their priorities; the BSs that already have carrier reservation during PPR-CH do not broadcast their priorities, and remain silent. All the BSs know that the PPR-CH signals are broadcast every period of R slots. Figure 3 illustrates the QDCR superframe and its channels.

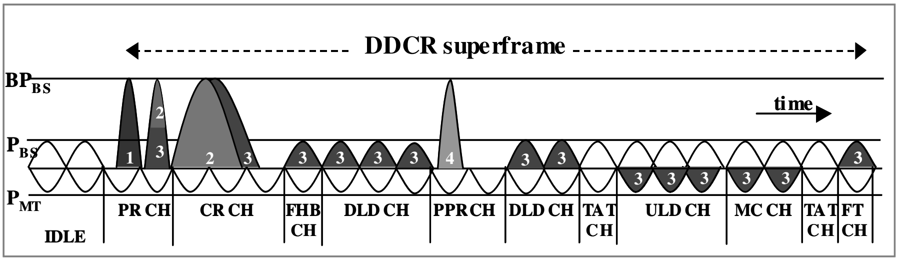


Fig. 3. The channels of the QDCR superframe. In the figure, three mutual interfering BSs compete for one carrier.

3.3. QDCR Carrier Selection

Each BS maintains and constantly updates the values of a parameter list named Congestion Factor (CF). This parameter indicates the number of BS's neighbors that compete or use each carrier during a recent period The notation $CF_{i,r}$ denotes the CF maintained by the BS_i for carrier F_r ($0 \le r < M$). $CF_{i,r}$ is updated according the rules:

1. When BS_i selects a carrier F_r it sets $CF_{i,r}=1$.
2. When BS_i realizes that F_r is reserved by another BS it sets $CF_{i,r}:=CF_{i,r}+1$.
3. It is possible for a BS_i, listening on a carrier F_r, to sense two or more signals bursts, which are transmitted simultaneously on p or c minislots by two or more BSs using the carrier F_r. In such case, the BS_i may not be able to receive useful information due to congestion, and it sets $CF_{i,r}:=CF_{i,r}+2$.
4. If during PR-CH or PPR-CH on a carrier F_r, R PBS signals are broadcast on p-slots of different order of the p-slot used by BS_i, if R_x of them are the only PBS signals per p-slot then $CF_{i,r}:=CF_{i,r}+R_x+2*(R-R_x)$, according to rule 3.
5. If BS_i loses a competition for carrier F_r, it sets $CF_{i,r}:=CF_{i,r}+2$ if for at least one c-slot receives useless information due to congestion, otherwise, it sets it sets $CF_{i,r}:=CF_{i,r}+1$.

When choosing a carrier, the BS should choose the carrier illustrating the less congestion. A BS_i keeps its Selection Parameters (SP_i) list, as follows:

$$SP_{i,r}=(CurrentTime_i\text{-}LastVisitTime_{i,r}+1)/CF_{i,r} \quad 0\le r<M,$$

A BS_I selects the carrier illustrating the minimum $SP_{i,r}$ value. The Current Time factor represents the time a BS takes a carrier selection decision. The Last Visit Time factor represents the last time the BS_i was using the carrier F_r. The rate of CF, diminishes in value with time, because the CF represents the congestion, i.e., the number of BSs, on a carrier during a recent time period. Thus, an ageing threshold is used to represent the uncertainty of the congestion for a carrier visited in the distant past. We introduce the following threshold:

$$CurrentTime\text{-}LastVisitTime_{i,r}>M*MeanFrameSize_i \quad 0\le r<M$$

A BS_i checks the time elapsed from the latest time the carrier F_r was used. If for a carrier F_r the ageing threshold is exceeded, the BS_i sets the $CF_{i,r}$ to a predefined value. Different predefined values can be applied. Results presented in [11] illustrate that the best policy for the predefined value, when the ageing threshold of BSi for the carrier F_r is violated, is to set the $CF_{i,r}$ to a value that is the lowest of the existing $CF_{i,k}$ values, i.e., $CF_{i,r}=\min(CF_{i,k})$, k<M, k≠r.

4. Simulation Environment and Results

To evaluate the QDCR performance, simulations were performed using the OPNET Simulator [12]. The carrier speed was set to 20Mbits/sec, and each TDMA slot was set to 54 bytes long. Each TDMA slot accommodates one MPDU. Each TDMA time frame was assumed to contain 5 slots for control information, i.e., for FH-CH, FT-CH, TAT-CH, and MC-CH. For the simulations, we used a combination of both real time (*rt*) and non real time (*nrt*) connections. *rt* connections classified to constant rate *rt* (*crt*), and variable rate *rt* connections (*vrt*). Each *crt* connection is simulated by a periodic MPDU generator. We assumed 64Kbps *crt* connections. For a 20Mb/sec channel, each 64kbps *crt* connection produces one MPDU every 260 time slots, approximately. We used 50 *crt* sources (25 uplink and 25 downlink) per BS. On the other hand, each *vrt* or *nrt* source is modeled by a Discrete time Batch Markov Arrival Process (D-BMAP). For such a process we consider the TDMA slot as the time unit. According to [13], the traffic load produced by a source can be approximated by the super-position of U equivalent ON/OFF minisources. For each *vrt* or *nrt* source, m was set to 256Kbps and s^2 was set to 128Kbps, where m and s^2 are the mean and the variance of the transmission rate, respectively, whilst the parameter U was set to 10 [13]. For the rt connections (i.e., *crt* and *vrt*) the MPDU Transfer Delay (MTD) was set to 50 time slots (i.e., approximately 1 msec), and the Di^{th} (eq. 1) parameter was set to 10^{-6}. For the *nrt* connections the MTD was set to 100 time slots (i.e., 2msec), and the Li^{th} was set to 10^{-6}. Each BS assumed to contain a cumulative buffer of 200 MPDUs. We considered two different load classes, load class 1 and 2. For load class 1 each BS accommodates 10 *vrt*, 10 *nrt*, and 30 *crt* connections. For lad class 2 each BS accommodates 20 *vrt*, 20 *nrt*, and 30 *crt* connections. The simulation duration was set to 7 days (approximately for $2{*}10^{10}$ TDMA time slots). For the QDCR mechanism, overheads are due to control channels (PR-CH, CR-CH, PPR-CH). In the simulations we have used two slots for each of the PR-CH and PPR-CH, with 10 p-slots in total, corresponding to 10 discrete priority reservation values. On the other hand, CR-CH resolves contention with 5 c-slots per contention slot, and a granularity factor (g) of 0.1. To simulate the interference environment we assumed that all the activated BSs are interfere mutually. QDCR mechanism was compared with alternative carrier selection mechanisms such as Random Choice (RC) and Round Robin (RR). In RC the BS selects randomly one carrier to compete for its reservation. In RR, once the BS starts its operation, it randomly selects a carrier F_r ($0 \leq r < M$) to compete for. In the next selection, the BS chooses the next carrier (i.e., F_{r+1}, if r<M-1, or F_0 otherwise). We have performed simulations using different combinations of operating BSs (N) and available carriers (M). Figures 4, 5, and 6 illustrate the mean reservation delay (in time slots) when 8

BSs compete to reserve 4 available carriers, for LJF, LDJF and DJF competing disciplines, when load class 1 is used. For QDCR it is assumed that PPR-CH is every 20 slots (parameter R).

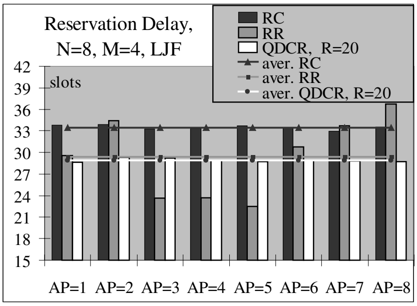


Fig. 4. Reservation delay of the LJF, for RR, RC, and QDCR (load class=1).

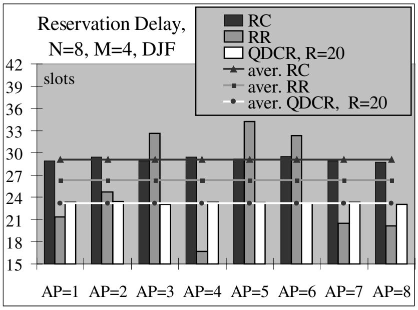


Fig. 5 Reservation delay of the DJF, for RR, RC, and QDCR (load class=1).

Figures 4, 5, and 6 illustrate that the DJF discipline achieves the lower reservation delays among all the alternative disciplines. QDCR achieves lower reservation delays, than the RC and the RR disciplines. Furthermore, even if the averaged, among the activated BSs, reservation delay for QDCR and RR are almost equal, the QDCR can guarantee fair reservation delays for the operational BSs, whilst RR does not. These observations are valid for all the contention disciplines. During the simulations we have observed that the QDCR mechanism achieves 20-40% lower reservation delays, than the RC or the RR disciplines, depending on the operational BSs (N), the available carriers (M), and the offered traffic load per BS. Figure 7 compares the performances of the LJF, DJF and LDJF disciplines. The better performance of the DJF, in terms of reservation delay, was confirmed for load class 2, as well.

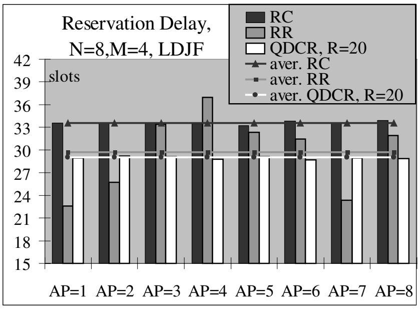


Fig. 6. Reservation delay of the LDJF, for RR, RC, and QDCR (load class=1).

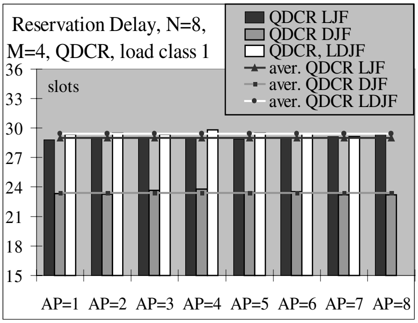


Fig. 7. Comparison of LJF, DJF, LDJF, (QDCR, load class=1).

Another parameter measured in the simulations is the success when predicting carriers' congestion levels. The CF, and the SP parameter lists, as well as the PPR-CH, are used by BSs to estimate the congestion level on carriers. Each BS attempts to predict carriers congestion levels when it selects a carrier to compete for; that is, at the end of a reservation, when losing on a competition, or when it realizes, through

PPR-CH, that another BS with higher priority is sensing the same carrier. Thus, success is the case where by, the carrier selected by the BS accommodates a minimum number of BSs compared to other carriers containing an equal or larger number of BSs at the same time.

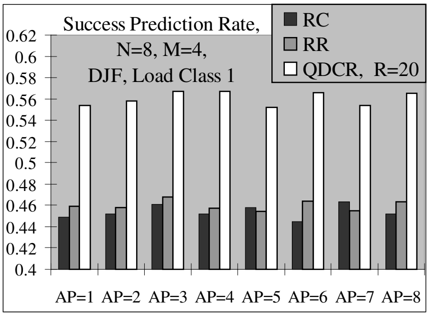


Fig. 8. Success prediction rate for RR, RC and QDCR (load class=1, M=4, N=8)

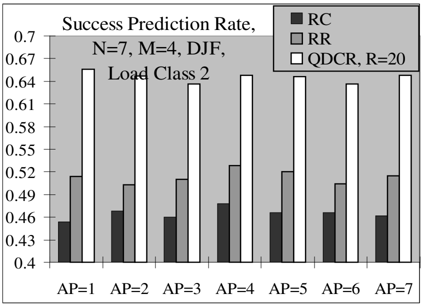
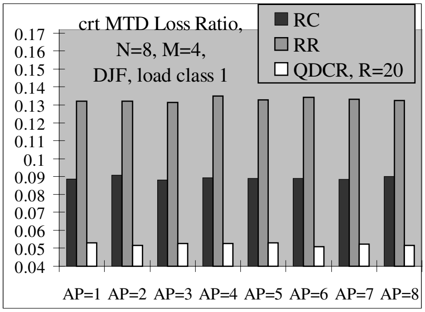


Fig. 9. Success prediction rate for RR, RC and QDCR (load class=2, M=4 N=7)

Figures 8, and 9 illustrate the prediction success rate of the QDCR, RR, and RC methods, for load class 1 (N=8, M=4), and 2 (N=7, M=4), respectively, when DJF competition discipline is used. For the QDCR, we have used R=20 (the PPR-CH is repeated every 20 slots). As figures 8 and 9 depict, the QDCR method achieves to predict carriers congestion levels with accuracy of 55% (load class 1), and 65% (load class 2). This difference is expected, since we have used the same value for R parameter (R=20) for both load classes. The mean TDMA frame size is equal to 15.2, and 50.8 slots, for load class 1, and 2, respectively; thus, for load class 1 the carriers' status changes more dynamically, than for load class 2. Setting R=20, for load class 1, the PPR-CH are repeated every two TMDA frames, whilst, for load class 2, the PPR-CH are repeated twice within one TDMA frame. QDCR achieves better success prediction rates for load class 2, because the BSs could switch carrier more times per frame, in a steadier environment, than in the case of load class 1. QDCR can achieve better prediction results, if period R is low, but there is a tradeoff between overheads (due to control information introduced to achieve higher success rate), and the reservation delay. When R=20 we observed that the overheads due to PPR-CH are 5%, thus QDCR is expected to introduce 5% more control information, than RR or RC disciplines, increasing the reservation delay by 5%. The buffer management policy that we have used is as follows. In every slot, each BS checks for MTD violations, and drops *crt*, *vrt*, and *nrt* MPDUs, in this order. If new MPDUs arrived, we time-tag them, and then if the buffer can accommodate these MPDUs, we place these MPDUs in the buffer. Otherwise, we drop from the buffer an equal number of aged MPDUs, in order to accommodate the new load. Concerning the buffer dropping policy, we first drop the older *nrt*, *vrt*, and *crt* MPDUs, in this order. This policy was used for all the set of the simulations. For N=8 operational BSs, M=4 available carriers, and for load class 1, we have observed that there are no buffer overflow conditions for the QDCR, RC or RR disciplines. This is due to the buffer management policy, since the algorithm drops MTD violated MPDUs first. On the other hand, we have observed *crt* and *vrt* MTD violations, illustrated in figures 10 and 11. From these figures it is concluded that QDCR produces low dropping ratios of *rt* MTD violated

MPDUs, achieving a reasonable MPDU loss ratio, equal $5*10^{-5}$, for both *crt* and *crt* type of connections.



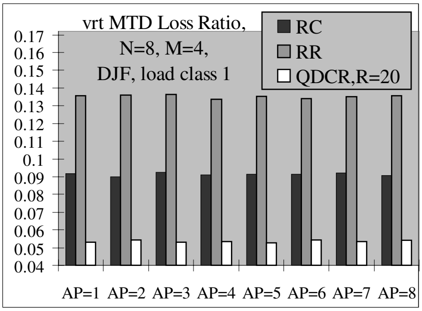
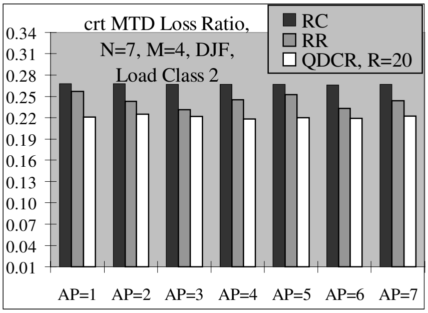


Fig. 10. *crt* MPDUs Loss Ratio due to MPDU Delay violations (load class 1)

Fig. 11. *vrt* MPDUs Loss Ratio due to MPDU Delay violations (load class 1)

For N=7 BSs, M=4 carriers, and load class 2, figures 12, 13, and 14 illustrate *crt*, *vrt*, and *nrt* MPDUs Loss Ratio, respectively, due to MTD violations, for QDCR, RC, and RR methods, whilst figure 15 depicts the cumulative rt MPDUs Loss Ratio.. Figures 12, 13, 14 and 15 illustrate that QDCR achieves lower MPDU losses on the contenting environment of the load class 2, than the disciplines RR or RC. Even if the MTD thresholds are not satisfied, due to the absence of a connection admission policy, QDCR achieves fairness among the competing BSs, concerning the MTD



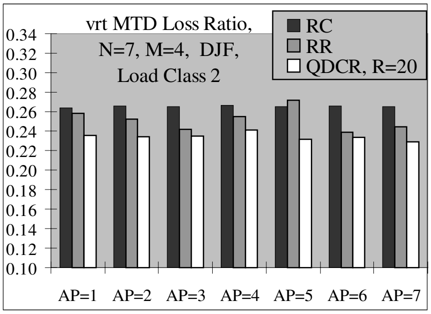


Fig. 12. *crt* MPDUs Loss Ratio due to MPDU Delay violations (load class 2)

Fig. 13. vrt MPDUs Loss Ratio due to MPDU Delay violations (load class 2)

Figure 16 depicts the MPDU dropping ratio due to buffer overflow conditions, for all the types of the connections. Figure 16 shows that QDCR produces low dropping ratios due to buffer overflow, achieving a reasonable MPDU loss ratio, less than $3*10^{-6}$, for all the types of all kinds of MTs connections. In figure 17 the sum of MPDU losses is illustrated. Figure 17 takes into account all the types of losses (Transfer Delay Violations, Buffer Overflows) for all kinds of connections. We can observe that the QDCR schedules the heavy traffic in a fair and more resourceful fashion, producing lower dropping ratios than RR or RC disciplines.

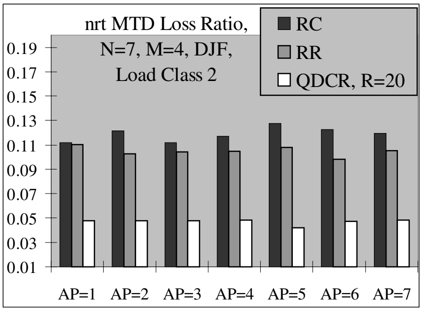


Fig. 14. n*rt* MPDUs Loss Ratio due to MPDU Delay violations (load class 2)

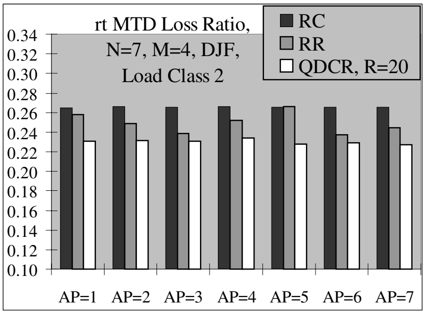


Fig. 15. *rt* MPDUs Loss Ratio due to MPDU Delay violations (load class 2)

5. Concluding Remarks

We have introduced QoS based competition rules for a distributed reservation method, which applies to interfering ATM or IP capable BSs competing for reservation in an unlicensed multicarrier wireless environment. The introduced QDCR mechanism takes into account the differentiated level of service required by the MTs in order to apply a priority based resource reservation method. Moreover, based on real time measurements of carriers' congestion levels, the QDCR mechanism assists BSs to select the least congested carrier in order to compete for its reservation. The QDCR mechanism is immune to topology changes, does not increase power consumption on MTs, and, finally, requires no frequency preplanning.. Furthermore, QDCR imposes no limit on the number of BSs operating in a common area, or on the number of neighboring BSs. The QDCR mechanism achieves to allocate shared resources efficiently. In order to be more virtue for the level of offered service to various types of connections, with different requirements in terms of loss and delay, the QDCR mechanism should be combined with a distributed wireless CAC. The latter could take into account the QDCR decisions, determine if the system is under heavy load conditions, and regulate the admission policy, accordingly.

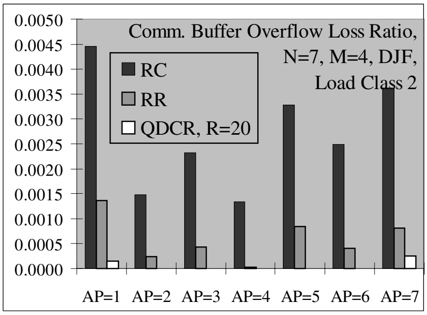


Fig. 16. Loss Ratio due to Buffer Overflow (load class 2)

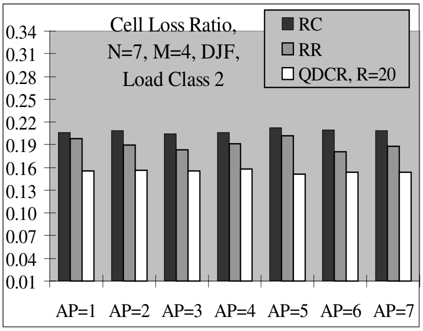


Fig. 17. Loss Ratio due to MTD and Buffer Overflow (load class 2)

References

[1] Braden, R., Clark, D., Shenker, S., “Integrated Services in the Internet Architecture: An Overview”, IETF RFC 1633, 1994.

[2] Blake, S., Black, D., Carlson, M., Davies, E., Wang, Zh., Weiss, W., “An architecture for Differentiated Services”, IETF RFC 2475, 1998.

[3] ETSI, "High Performance Radio Local Area Network (HIPERLAN) Type 2; Requirement and architectures for wireless broadband access", TR 101-031, 1999

[4] V. Li, and X. Qiu, "Personal communications systems," Proc. of the IEEE, Sept. ‘95

[5] M. Frodigh, “Bounds on the performance of DCA algorithms in highway microcellular systems,” IEEE Trans. Vehic. Tech., vol. 43, no. 3, Aug. 1994.

[6] Heinanen, J., et. al., “Assured Forwarding PHB group”, RFC 2597, 1999.

[7] H. Jacobson, V. Nichols, K. Poduri, “An Expedited Forwarding PHB”, RFC 2598, 1999.

[8] N. Passas, L. Merakos, S. Paskalis, D. Vali, “Quality-of-Service-Oriented medium access control for wireless ATM networks” November 1997 issue of IEEE Comm. Mag.

[9] G. F. Marias, D. Skyrianoglou, and L. Merakos, “A Centralized Approach to Dynamic Channel Assignment in Wireless LANs,” Proc. IEEE INFOCOM99, NY, USA, Mar. 1999.

[10] ACTS Project, “The Magic WAND”, Deliverable 3D6, “Wand DCA Scheme," CEC Del. AC085/INT/ACT/DS/P/034/b1, Aug. 1998.

[11] G. F. Marias, and L. Merakos, “Performance Estimation of a Decentralized Mutlicarrier Access Framework In Unlicensed Wireless Systems,” Proc. ICC2001, June 2001, Finland.

[12] OPNET Modeler, MIL 3, Inc., 1993.

[13] C. Blondia, and O. Casals, “Performance analysis of statistical multiplexing of VBR sources,” Proc. INFOCOM ’92.

A Simple Model for Calculating SIP Signalling Flows in 3GPP IP Multimedia Subsystems

Alexander A. Kist and Richard J. Harris

RMIT University, BOX 2476V, Victoria 3001, Australia
{kist,richard}@catt.rmit.edu.au
http://www.catt.rmit.edu.au

Abstract. The 3rd Generation Partnership Project (3GPP) uses the IETF Session Initiation Protocol (SIP) as a signalling protocol in the IP Multimedia Subsystem for 3rd generation UMTS networks. Signalling Messages sent using the SIP protocol pass through intermediate SIP nodes in such a way that the message size grows as a result of additional data being added to the message. Modelling of network flows in this case requires careful attention. A simple flow model is presented to get an appreciation for the size of the expected signalling flows. This is particularly important for developing dimensioning models for SIP in 3GPP networks and for the further investigation of Quality of Service (QoS) mechanisms to provide resources needed for signalling. A methodology is presented which defines the minimum requirements for the bit error rate on links used by signalling traffic.

1 Introduction

The 3rd Generation Partnership Project (3GPP) [1] is a global initiative to develop technical specifications for 3rd Generation Mobile Systems. 3GPP has decided to use the IETF Session Initiation Protocol (SIP) as the signalling protocol for the IP Multimedia Subsystem. The standardisation process for both 3GPP and SIP is currently in progress. This paper introduces a simple model to compute the flows on links between SIP nodes and is based on ideas involving feedback systems.

The Session Initiation Protocol (SIP) is a client-server protocol and used as a signalling protocol in IP environments. It performs user location, session establishment, session management and participant invocation. The SIP protocol is defined in RFC 2543 [2] and the new version of the specification is discussed as an Internet Draft [3] (work in progress). There are several publications available that provide an introduction to the SIP protocol. Examples include works by [4] Rosenberg and Schulzrinne/Rosenberg [5].

Using the SIP protocol on a large scale in 3GPP networks requires Quality of Service (QoS) observations for the signalling to be able to guarantee QoS standards for customers and optimise the network performance [6]. Unlike traditional Signalling System No. 7 (SS7) networks the SIP protocol can use the same

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 924–935, 2002.

transport network as the voice bearer and other services - the underlying IP network[1]. Several QoS mechanisms have been proposed for use in IP networks (e.g. Xiao [7]). To apply these mechanisms in an appropriate way, an understanding of the flows in the network is required.

Furthermore SIP was designed for the Internet environment but operator networks are different in many ways. Operators require more control over their network, billing and accounting issues and a certain quality is required based on contracts with customers. Several publications concerning the SIP protocol consider only a few intermediate proxies (e.g. Eyers [8]). To satisfy the specific needs of operators standard call flows consist of seven intermediate proxies or more [9].

3GPP uses several different SIP proxy servers. They are abbreviated by *CSCF*, the *Call Session Control Function.* These different signalling nodes serve various functions such as database request, recording state information for billing, and serve as hiding nodes. These details are not of a specific interest in this paper and the nodes are seen as general SIP proxy servers. More details on the specific functions can be found in the technical specifications [9].

For modelling purposes SIP requests can be divided into requests that use a hop-by-hop reliable mechanism (e.g. INVITE, CANCEL) and requests that use an end-to-end reliable mechanism. In the former, the model introduced in this paper has to be applied for one hop only, for the latter the model has to be applied end-to-end. For this modelling approach it is assumed that all proxies are statefull. Furthermore it is assumed that the RTT is smaller than the SIP timer, since otherwise messages are resent due to timeout and not to loss.

Section 5 formulates a model to describe the flows on a single SIP connection. This model requires the lost message model discussed in Section 2, the message loss probability discussed in Section 3 and the model for changing message sizes presented in Section 4 as inputs. Section 6 discusses possible simplifications and the calculation of a bit error boundary. Section 7 discusses the results of the application of this model and illustrates them graphically. The paper concludes with a discussion of further work and additional remarks.

2 Lost Message Model

The SIP protocol is transport protocol independent. Since the only mandatory transport protocol for SIP is UDP, it needs to incorporate its own end-to-end reliability mechanism. In particular the SIP extension[2] known as "Reliability of Provisional Responses" introduces an additional reliability mechanism for the provisional response in a SIP call flow, which is used by 3GPP. This section discusses a flow model that takes the reliability mechanism into account. The

[1] Future SS7 networks can also run over IP transport networks possibly using the Stream Control Transmission Protocol (SCTP). In this context similar QoS consideration are required.

[2] The extension is now part of the Internet Draft [3]

modelling approach of this behaviour is based on the model for repeated attempts in [10].

A message flow[3] M between two nodes has to be transmitted over a link that is assumed to have an error probability $P_E(M)$[4]. This link has to accommodate the original message flow M. Consequently, a flow of $(M \cdot P_E(M))$ will be lost on the link due to the message error and has to be retransmitted on this same link. This new flow $(M \cdot P_E(M))$ is subjected to loss once again with probability $P_E(M)$. So the lost flow in this instance is then $(M \cdot P_E(M) \cdot P_E(M))$. If a message is resent n times this yields Equation (1), where F is the total flow on the link.

$$F = M + M \cdot P_E(M) + M \cdot P_E(M)^2 + \ldots + M \cdot P_E(M)^n \ . \tag{1}$$

This well-known geometric series can be summed as shown in Equation (2).

$$F = M \frac{1 - P_E(M)^{n+1}}{1 - P_E(M)} \ . \tag{2}$$

For an infinite number of retransmissions a simplification of Equation (2) is possible.

$$F = \lim_{n \to \infty} M \frac{1 - P_E(M)^{n+1}}{1 - P_E(M)} = \frac{M}{1 - P_E(M)} = M + \frac{M \cdot P_E(M)}{1 - P_E(M)} \ . \tag{3}$$

The SIP protocol specifies that the messages are resent with a maximum number of reattempts[5] $n = 7$. This is usually implemented by a timer rather than counting re-transmissions. Under certain conditions the number of reattempts can be reduced to n=4. A message that is lost n times will cause a termination of the connection. Since the error is very small[6] the formula for the limiting case is used for simplicity.

In SIP some messages depend on other messages. If an upstream[7] message M_2 (eg. the message 200OK) in the SIP protocol is lost it forces the resending of downstream messages M_1 (eg. PRACK). $P_E(M_1)$ is the probability that message M_1 is lost on the downstream path and $P_E(M_2)$ is the probability that message M_2 is lost on the upstream path. The loss of message M_1 causes the "time out" of the sender and triggers the resending of the message. This case is covered by Equation (4).

$$F_D(M_1) = \frac{M_1}{1 - P_E(M_1)} \ . \tag{4}$$

If the M_2 message corresponding to message M_1 is lost there is no mechanism to recognise this loss. It is resent as a response to a newly sent message M_1. A loss

[3] A message flow is the number of bytes per time unit.

[4] For this approach it is assumed that the SIP messages are not fragmented.

[5] See [2] for details.

[6] For an extremely high bit error rate of $P_E = 10^{-2}$ the error is $1 - P_E(M)^5 = 1 - 10^{-10}$.

[7] Upstream is in SIP the direction from the server to the client.

of message M_2 therefore causes an additional resent message M_1. Equation (5) describes the upstream flow $F_U(M_2)$ and Equation (6) describes the downstream flow $F_D(M_2)$ caused by a lost message M_2.

$$F_U(M_2) = \frac{M_2}{1 - P_E(M_2)} \ . \tag{5}$$

$$F_D(M_2) = \left(\frac{M_1}{1 - P_E(M_2)} - M_1 \right) \frac{1}{1 - P_E(M_1)} \ . \tag{6}$$

Note that the subtraction of M_1 in Equation (6) is due to the fact that this equation only covers additional flows of M_1 and not the original message. The factor is due to the possibility of loss of this additional flow. Equation (6) yields (7):

$$F_D(M_2) = \frac{M_1 \cdot P_E(M_2)}{(1 - P_E(M_2))(1 - P_E(M_1))} \ . \tag{7}$$

If both flows $F_D(M_1)$ and $F_D(M_2)$ are taken into account, this yields Equation (8).

$$F_D = \frac{M_1}{(1 - P_E(M_2))(1 - P_E(M_1))} \ . \tag{8}$$

This equation shows the expected result for the overall flow F_D. It states that the original message flow M_1 is increased by the probability that message M_1 is lost on the downstream path and the probability that message M_2 is lost on the upstream path. The next section discusses the message loss probability and how it is determined by the bit error rate on the underlying link.

3 Message Loss Probability and Bit Error

The reliability of a communication link can be described by the *Bit Error Ratio* (BER). It states that *BER*% of all transmitted bits are corrupt due to transmission errors. The following section discusses the probability that a message of size m bytes sent on a link with a particular BER is corrupt.

A message is lost if one or more bits of the message are corrupt. The probability that a bit error occurs is *BER*. The probability P_E that a message is corrupt can be calculated with the binominal distribution. For a message with the size of M bytes this yields Equation (9).

$$P_E(M) = \sum_{k=1}^{8M} \binom{8M}{k} BER^k (1 - BER)^{8M-k} \ . \tag{9}$$

Because the message size is a byte value, the factor 8 calculates the message size in bits. The probability $\overline{P_E(M)}$ that a message is not corrupt can be calculated with Equation (10).

$$\overline{P_E(M)} = 1 - P_E(M) \ . \tag{10}$$

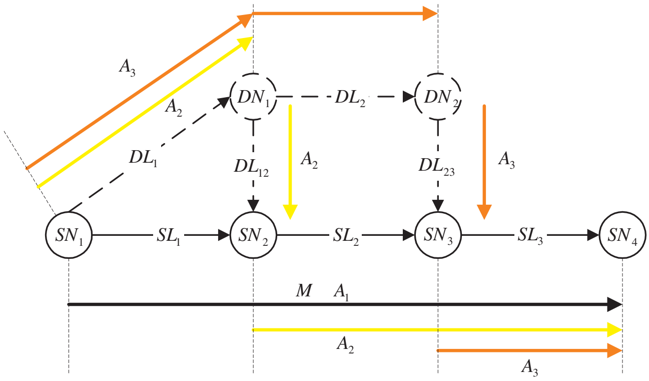


Fig. 1. Transformed Network Example

The $k = 0$ term in the sum in Equation (9) describes the probability that no bits are corrupted. Thus, Equation (9) may be further simplified using Equation (10) to yield the message loss probability.

$$P_E(M) = 1 - (1 - BER)^{8M} \quad . \tag{11}$$

The next section discusses the effects of changing the message size on network flows.

4 Model for Changing Message Size

A characteristic of the SIP protocol is that the message size changes at every node through which it passes. This is due to the fact that every node inserts its own DNS address in the SIP message *VIA* header for a downstream message and removes its address from an upstream message[8]. Comparing this behaviour with other flow models this is rather different from conventional flow models. A flow observed in one part of the network has a certain size but in other parts of the network the flow has a different size. This behaviour especially violates the normal conservation of flow property of flow models. This section describes a model that enables the conservation of flow for a network with changing message sizes. Firstly, a model for increasing message sizes for downstream flows is discussed.

The following discussion uses an example network with four nodes. The nodes SN_1 to SN_4 are connected using three links SL_1 to SL_3. A message is sent on the links from the origin SN_1 to the destination SN_4. A message sent on a link can be lost with an error probability P_E(Message Size). At every node the message is increased in size by a constant term A. This is true in SIP for requests sent from the client to the server. Node SN_1 can be seen as the client and node

[8] The *route* field cause the same problem. See the SIP specifications [2] for more detail.

SN_4 as the server. To calculate the flows in this network a transformed network is defined. It is depicted in Figure 1. Two additional dummy nodes DN_1 and DN_2 corresponding to the original intermediate nodes are inserted. The nodes are connected with four additional dummy links DL_1, DL_2, DL_{12} and DL_{23}. The dummy links DL_{12} and DL_{23} have a zero message loss probability and the links DL_1 and DL_2 have a message error probability that is corresponding to the original links SL_1 and SL_2 respectively.

Node SN_1 generates the flows for the original message $OM + A_1$ on link SL_1 and the flow for A_2 and A_3 on link DL_1. Additionally, the flows corresponding to the resending of the lost messages are induced in the network. Link SL_1 (Equation (12)) accommodates therefore the original flow $OM + A_1$, the flow that is lost on this link (second term), the flow that will be lost on SL_2 increased by the loss on link SL_1 (third term) and the flow which will be lost on SL_3 increased by the loss on link SL_2 and SL_1 (fourth term).

$$\begin{aligned} F(SL_1) = (OM + A_1)\Big(1 + \tfrac{P_E(SL_1)}{1-P_E(SL_1)} + \tfrac{P_E(SL_2)}{(1-P_E(SL_1))(1-P_E(SL_2))} \\ + \tfrac{P_E(SL_3)}{(1-P_E(SL_1))(1-P_E(SL_2))(1-P_E(SL_3))}\Big) \ . \qquad (12) \end{aligned}$$

Dummy link DL_1 has to accommodate similar flows for A_2 and A_3. Link SL_2 (Equation (13)) has to accommodate the original flow $OM + A_1 + A_2$, the additional resent flow that will be lost on link SL_2 (second term) and the flows that will be lost on link SL_3 increased by the loss on link SL_2 (third term).

$$F(SL_2) = (OM + A_1 + A_2)\Big(1 + \tfrac{P_E(SL_2)}{1-P_E(SL_2)} + \tfrac{P_E(SL_3)}{(1-P_E(SL_2))(1-P_E(SL_3))}\Big) \ . \qquad (13)$$

Dummy link DL_2 carries a similar flow for A_3. Link SL_3 finally carries the original flow $OM + A_1 + A_2 + A_3$ as well as the flow that is lost on this link (Equation (14)).

$$F(SL_3) = (OM + A_1 + A_2 + A_3) \cdot \left(1 + \frac{P_E(SL_3)}{1 - P_E(SL_3)}\right) \ . \qquad (14)$$

The message error probabilities P_E are functions of the original message size on the links (Equation Set (15)).

$$\begin{aligned} P_E(SL_1) &= f(OM + A_1) \\ P_E(SL_2) &= f(OM + A_1 + A_2) \\ P_E(SL_3) &= f(OM + A_1 + A_2 + A_3) \ . \end{aligned} \qquad (15)$$

As the message size increases for requests it decreases for responses. The network in Figure 1 is used for the discussion as well. It requires the additional reverse links rSL_1, rSL_2 and rSL_3 with the corresponding errors $P_E(rSL_1)$, $P_E(rSL_2)$ and $P_E(rSL_3)$ respectively[9]. The flow on the reverse link rSL_3 consists

[9] Leading indices r are used to indicate parameters for the reverse direction, the response direction.

of the original reverse message rOM and the terms $A_1 + A_2 + A_3$. The flow on the link including the terms for the lost messages is depicted in Equation (16).

$$F(rSL_3) = (rOM + A_1 + A_2 + A_3) \cdot \left(1 + \tfrac{P_E(rSL_3)}{1-P_E(rSL_3)}\right) \ . \tag{16}$$

Link rSL_2 accommodates the original reverse flows $rOM + A_1 + A_2$ and the flow for the messages lost on link rSL_2. Similar observations for link rSL_1 require the original flow $rOM + A_1$ as well as the flow for the lost messages on rSL_1. As above, the message error probability depends on the size of the message on the link. Theses dependencies are similar to Equation Set (15). The next section formulates the flows on the links in SIP connections.

5 Calculating the Flows

Using the models from the previous sections it is possible to formulate the flows for the SIP connection. In this section, the following notation is used: The connection consists of SIP nodes SN_1 to SN_{max}. Every node adds a value of A_{SN} bytes to the message. The original message is of size OM. Unidirectional SIP links SL with bit error BER_{SL} connect the nodes. Where link SL_1 emanates from SN_1 and terminates at SN_2. The size of a message on link SL can be calculated using Equation (17).

$$M(SL) = OM + \sum_{n=1}^{SL} A_n \ . \tag{17}$$

It should be noted that for exact practical calculations, the size of the UDP header has to be added as well. A message consists of the original message part OM and a number of terms A_n.

Knowing the message size on the links it is possible to calculate the message loss probability $P_E(SL)$. Equation (17) shows this calculation:

$$P_E(SL) = 1 - (1 - \mathrm{BER}_{SL})^{8M(SL)} \ . \tag{18}$$

With the message size and the message error probabilities known, all input parameters are available to apply the increasing message model of Section 4. Equation (19) calculates the flows on link SL for a downstream message (request).

$$FL(SL) = M(SL)\left(1 + \sum_{m=SL}^{SL_{max}} \frac{P_E(m)}{\prod_{n=SL}^{m}(1 - P_E(n))}\right) \ . \tag{19}$$

The flows for a response in the reverse direction are calculated by Equation (20) in a similar way. The reverse message size rOM and the bit error probabilities for the reverse links are required.

$$FL(rSL) = \frac{M(rSL)}{(1 - P_E(rSL))} \ . \tag{20}$$

If one of the message flows depends on other messages, the appropriate flow has to be increased by the overall message error probability for the corresponding direction (Equation (21)).

$$FL = FL(SL)\frac{1}{\prod_{n=SL}^{SL_{max}}(1 - P_E(rn))} \ . \qquad (21)$$

The following section discusses possible simplifications.

6 Simplifications

For the special case of equal bit errors on all links[10], certain simplifications are possible. It can be shown that the flows have a linear dependence on the bit error rate, if the bit error rate is small enough. Equation (22) shows Equation (19) with a single bit error rate value BER.

$$FL(SL) = M(SL)\Big(1 + \sum_{m=SL}^{SL_{max}} \frac{1-(1-\mathrm{BER})^{8M(m)}}{(1-\mathrm{BER})^{s(m)})}\Big) \ . \qquad (22)$$

With:

$$s(m) = 8 \sum_{n=m}^{SL_{max}} M(n) \ . \qquad (23)$$

Simplifying the sum in Equation (22) yields Equation (24).

$$FL(SL) = M(SL)\Big(1 + \frac{1-(1-\mathrm{BER})^{s(SL)}}{(1-\mathrm{BER})^{s(SL)})}\Big) \ . \qquad (24)$$

Equation (24) can be further simplified to Equation (25).

$$FL(SL) = M(SL)(1-\mathrm{BER})^{-s(m)} \ . \qquad (25)$$

To show the linear dependence, Equation (25) is written as a MacLaurin series. For the first four terms this yields Equation (26) ($s = f(SL)$.

$$FL(SL) = 1 + s \cdot \mathrm{BER} + \frac{s^2+s}{2}\mathrm{BER}^2 + \frac{s^3+3s^2+2s}{6}\mathrm{BER}^3 + R \ . \qquad (26)$$

In order to define the region for which the linear term is a sufficient approximation and therefore Equation (27) is valid, Equation (28) calculates the fraction between the linear and the quadratic term.

$$FL(SL) = 1 + s \cdot \mathrm{BER} \ . \qquad (27)$$

[10] This case will not apply for 3GPP end-to-end connections since the bit error will be higher over wireless links at the network edge. For calculations between SIP proxies within homogenous networks these assumptions are possible.

$$\frac{s+1}{2}\mathrm{BER} < \alpha \text{ and since } s \gg 1\text{: } \mathrm{BER} = \frac{2\alpha}{s} \ . \tag{28}$$

If α is chosen, the boundary where the linear approximations no longer holds can be calculated. This also defines the minimum requirement for the bit error rate to achieve sufficient performance, since otherwise the flows caused by the resending of lost messages increases exponentially. The calculation has to be done for the first link since its flows increase by the largest amount in the case of message loss.

In the case of different bit error rates on the link, the link with the worst bit error rate appears to dominate the connection and the loss on the previous links (See Figure 4), because the simplifications in this section consider the worst case and, therefore, provide an upper bound for the bit error rate. The maximum value of the bit error introduced in this section can be used in this situation as well. The worst bit error rate within the network has to be smaller than the upper bound for the bit error. For dependent messages this applies for the reverse connection as well.

7 Results

This section discusses results that have been found by applying the above model. The presented results are preliminary with a focus on the influence of the parameters and the increasing flows due to the resending of lost messages. Common for all examples, is the underlying SIP connection. It consists of 9 nodes and 8 intermediate links, respectively. The bit error rate is the first parameter that is discussed.

In the following example a SIP downstream connection with an original message size of 300 bytes[11] is observed. In every node the message is increased by 40 Bytes. The x-axis in Figure 2 depicts the bit error rate and the y-axis shows the percentage by which the flows increase due to the resending of corrupted messages. Both use a logarithmic scale. The curves in the graph represent different hops. Hop number 1 is the link emanating from the origination node and hop number 8 is the link that terminates at the distant node. For this example, it is assumed that the bit error rate is equal on all links. The graph shows that for bit error rates over 10^{-5} the flows on the first links increase rapidly. For a reasonable result, e.g. flows are not increased by more than 10%, a bit error ratio better than 10^{-6} is required. Figure 3 shows curves for the increasing message size, calculated with the original equation and curves calculated with the linear approximation. The curves display the situation for the first and the last link respectively. The graph verifies the analytic results from Section 6. The boundary calculated with $\alpha = 0.01$ yields a bit error of $\mathrm{BER} = 6.5 \cdot 10^{-7}$ for the first link and $\mathrm{BER} = 4.0 \cdot 10^{-6}$ for the last link. These values are drawn as vertical

[11] For the the examples in this section, a message size of 300 bytes was chosen as a typical size of a SIP request without a SDP part and 40 bytes was chosen as a typical VIA-URL size. Different message size assumptions have no impact on the principle results, but further work will investigate the exact influence of the message size.

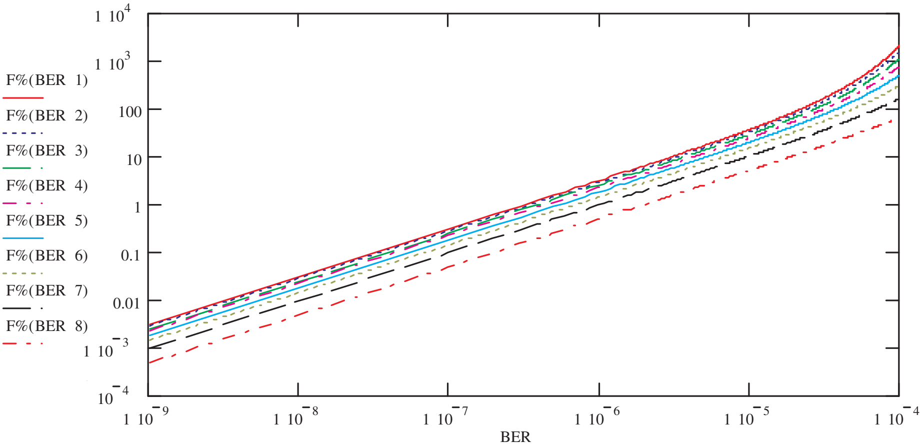


Fig. 2. Increasing Message Size versus Bit Error: All Links

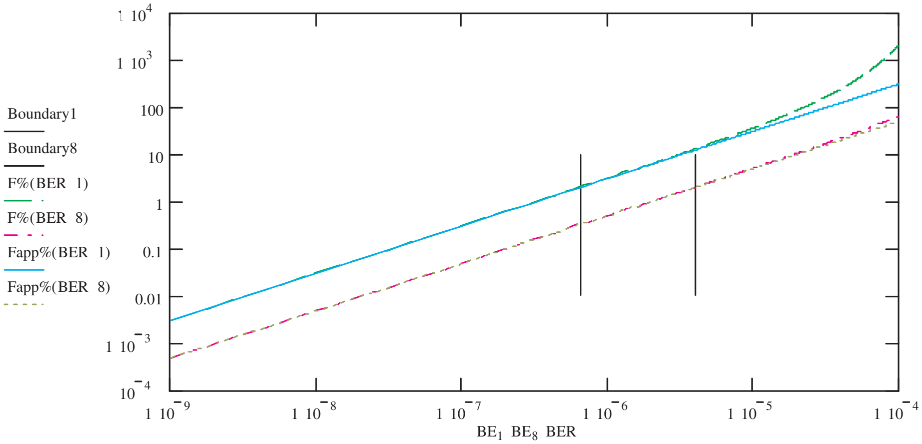


Fig. 3. Increasing Message Size versus Bit Error: Approximation

lines on the graph. In a practical case, the bit error requirements for the first link have to be used.

Figure 4 depicts a graph where one link in the connection has a worse bit error rate than the other links. The x-axis shows the node number and the y-axis shows the increase in flow expressed as a percentage of the original flow. All links but one have a bit error rate of 10^{-6}. The first curve shows the original case where all the links have the same bit error. For the second curve, the bit error of link 1 is set to 10^{-5}, for the third curve the bit error of link 2 is increased to 10^{-5}, for the fourth curve the bit error of link 5 is increased and for the last curve the last link has the worst bit error rate.

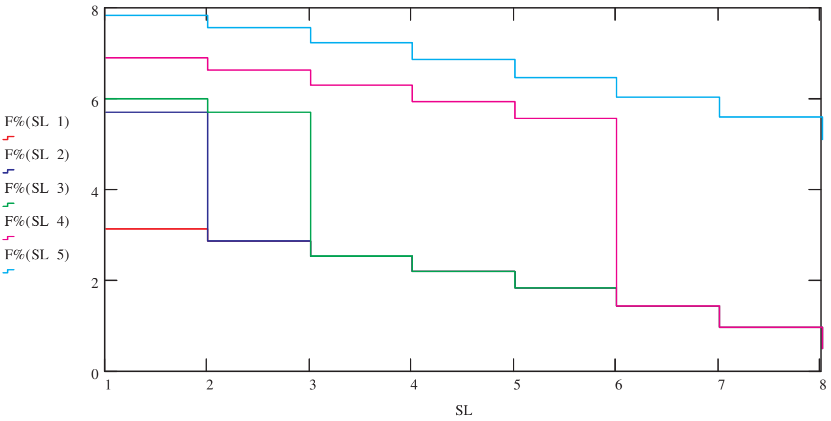


Fig. 4. Increasing Message Size on Different Links

The graph shows that links with higher BERs further downstream increase the flows on the previous links. The result that the flow is influenced to a greater extent by later links is caused by the fact that the message size increases downstream. If the message size had been constant, the curve for link 8 would cut the other curves.

The observation of the size of the message as a variable parameter shows that the increase of the message size is a linear function of the message size if the bit error rate is sufficiently small enough. The approximation of Section 6 also applies in this case since s in Equation (27) depends on the message size. This result is helpful if the message size is not a fixed value but a distribution of different message sizes. As long as the distribution is symmetrical around the average value, calculations done for an average message size provide a good result. The second expected conclusion is that for larger messages, a lower bit error rate is required to avoid unacceptably increased message flows. It can be shown that a boundary, similar to the one for the bit error, also exists for the message size where the dependencies are no longer linear. For practical message sizes and appropriate bit error rates, the linear assumption applies.

8 Further Work

This model describes the flows on one connection in 3GPP SIP based networks. It is intended to aid in the formulation of planning models that require the calculation of all accumulated flows in a 3GPP signalling domain. Such a model has to consider the different SIP message types in a call flow with their specific size and the signalling network structure of a 3GPP signalling network. Detailed investigations of message length and link error probability distributions are required. Methodologies to incorporate this modelling approach into an overall concept are introduced in [6]. It is also planned to formulate routing methodologies to optimise the signalling flows within 3GPP IP Multimedia Subsystems.

9 Conclusions

This paper has presented a methodology for calculating flows on SIP connections and has evaluated its performance. The model uses the bit error rate as the parameter which impacts on the message loss. In today's IP networks the bit error rate is considered to be of minor importance. But networks for mobile applications traditionally use a number of high bit error rate links, for example, microwave links at the edge of the network and the air interface connections to mobile equipment. These links possibly use link layer retransmission. This paper provides methodologies to enable qualified decisions whether link layer retransmission, due to high bit error rates, is required or not to operate the SIP signalling protocol on such links. The model can be easily adapted to consider the message loss due to overflowing queues in the network as well. The rationale for this paper was to provide an overall planning methodology to enable QoS for the signalling part in 3GPP IP Multimedia Subsystems.

Acknowledgements. The authors would like to thank Ericsson AsiaPacificLab Australia for their financial assistance for this work. The helpful comments by John Murphy and Bill Lloyd-Smith concerning the analytic results are gratefully acknowledged.

References

1. 3rd Generation Partnership Project: About 3GPP. `http://www.3gpp.org`. January 2001.
2. Handley, M., Schulzrinne, H., Schooler, E. and Rosenberg, J. D.: SIP: Session Initiation Protocol. RFC 2543 March 1999.
3. Rosenberg, Schulzrinne, Camarillo, Johnston, Peterson, Sparks, Handley and Schooler: SIP: Session Initiation Protocol. IETF Internet Draft <draft-ietf-sip-rfc2543bis-07.ps> (work in progress). February 2002.
4. Rosenberg, J. D. and Shockey, R.: The Session Initiation Protocol (SIP): A Key Component for Internet Telephony. Computer Telephony June 2000.
5. Schulzrinne, H. and Rosenberg, J. D.: The Session Initiation Protocol: Internet-Centric Signaling. IEEE Communications Magazine October 2000 134–141
6. Kist, A. A. and Harris, R. J.: QoS and SIP Signalling in 3GPP IP Multimedia Subsystems. Royal Melbourne Institute of Technology Melbourne, Australia, October 2001 (to appear).
7. Xiao, X. and Ni,L.: Internet QoS: A Big Picture. IEEE Network March/April 1999.
8. Eyers, T. and Schulzrinne, H.: Predicting Internet Telephony Call Setup Delay. In IPTel 2000 (First IP Telephony Workshop) April 2000.
9. 3rd Generation Partnership Project: IP Multimedia (IM) Subsystem - Stage 2 (Release 5). July 2001. (3GPP TS 23.228 V5.1.0)
10. Atov, I. and Harris, R.J.: A Mathematical Model for IP over ATM. IFIP-TC6 Networking Conference 2002, Pisa May 19-24 2002.

Dynamic Online Routing Algorithm for MPLS Traffic Engineering

W. Szeto, R. Boutaba, and Y. Iraqi

Department of Computer Science,
University of Waterloo,
200 University Avenue West,
Waterloo, ON, Canada, N2L 3G1
{wwszeto, rboutaba, iraqi}@bbcr.uwaterloo.ca

Abstract. The main contribution of this paper is a new online routing algorithm, called Dynamic Online Routing Algorithm (DORA), for dynamic setup of bandwidth guaranteed paths in MPLS networks. The goal of DORA is to accept as many network path setup requests as possible by carefully mapping paths with reserved bandwidth evenly across the network. The key operation behind DORA is to avoid routing over links that: 1) have high potential to be part of any other paths, and 2) have less residual bandwidth available. We compare DORA against other existing constraint-based routing algorithms based on two performance metrics: path setup rejection ratio and percentage of successful reroutes. Our result shows that DORA performs better than the other algorithms in both metrics.

Keywords: MPLS, Constraint-based Routing, Traffic Engineering

1 Introduction

We present a new online routing algorithm, called Dynamic Online Routing Algorithm (DORA), for construction of bandwidth guaranteed paths. DORA aims to utilize existing network resources and minimize network congestions by carefully mapping paths with specific bandwidth requirement evenly across the network topology. The main objective of DORA is to allow more path setups to be accepted into the network, and as a result, increased revenue for service providers.

The problem of establishing bandwidth guaranteed paths, as path setup request arrives one-by-one with no advance about future requests, has been studied elsewhere in [KL00,SWW01,FT00]. Our work is inspired by the Minimum Interference Routing Algorithm proposed in [KL00], but performs better in terms of request rejection ratio and rerouting percentage upon link/node failures with much less computation complexity. We will describe our algorithm in the context of MPLS-enabled networks. In a Multi-Protocol Label Switching (MPLS) network, incoming packets are assigned a label at the ingress node. Each packet follows a pre-computed path, which is identified by a set of labels, to reach its

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 936–946, 2002.

destination node. Constraint-based Routing (CR) extends MPLS path computation by ensuring that the resulting path satisfies a set of constraints. In this paper, we will consider path computation with bandwidth requirement as the constraint.

The organization for the rest of this paper is as follows: Section 2 identifies important design issues for routing algorithms in a MPLS network. In Section 3, we discuss some related works. Section 4 describes the details of DORA algorithm. Section 5 evaluates and compares DORA against other existing routing algorithms through network simulations. Section 6 summarizes the important points in the paper and proposes future work.

2 Design Issues of Constraint-Based Routing Algorithms

In this section, we briefly describe some of the important properties of a useful routing algorithm in the MPLS domain.

1. **Routing Constraint:** Constraints may include delay, jitter, loss ratio, bandwidth, administrative constraints and so on. It has been proven that finding the optimal route subjected to two or more additive and/or multiplicative metrics is NP-complete [GJ79]. In addition, it is generally difficult to obtain accurate values for certain metrics such as delay and jitter. In the rest of this paper, we will focus on routing algorithms that compute paths subjected to bandwidth requirement.

2. **Online Routing:** Offline constraint-based routing requires a demand matrix as input. A demand matrix describes the expected amount of data to be transmitted between a source-destination pair in the network at different times. In contrast, online constraint-based routing does not require a priori knowledge of the size and arrival time of each individual path setup request. This paper focuses on online constraint-based routing, as it is the appropriate approach to solving the dynamic path setup problem in MPLS networks.

3. **Computational Requirement:** Online routing algorithm compute paths as setup requests arrive at the network. In the case where the ingress node is operating in demand-driven mode, the path computation time is added to the overall response time that the user perceives. Therefore it is necessary for path computations to be as fast and as efficient as possible.

4. **Re-routing Performance:** Network topology changes are triggered by events such as link or node failure. When a link fails, all affected paths have to be re-routed to a different location in order to resume normal operations. Since each path is associated with reserved bandwidth, re-routing may fail due to insufficient residual resource. A good routing algorithm should carefully map network paths across the topology so that when a link fails, the chance of successfully rerouting affected paths is as high as possible.

5. **Link State Distribution:** The current standard link state advertisements (LSA) do not contain dynamic link information such as residual bandwidth and residual capacity. There have been work to extend the Interior Gateway Protocols (IGPs) to add dynamic link attributes to LSAs [KYK02,LS01]. We assume that the necessary bandwidth and topology related information used by the routing algorithm is available when needed.

3 Related Works

The most popular and widely used routing algorithm in MPLS networks is the shortest-path first algorithm (SPF) based on the number of hops. SPF selects the path that contains the fewest hops between the source and the destination node. One obvious problem with SPF is that it tends to route traffic onto the same set of links until these links' resource are exhausted. This leads to concentration of traffic on certain parts of the network. In addition, SPF typically accepts less path setups into the network than some other more advanced routing algorithms.

A more intelligent routing algorithm the Minimum Interference Routing Algorithm (MIRA) proposed in [KL00]. The objective of MIRA is to accept as many path setups into the network as possible by using the concept of critical links. Critical links have the property that when their capacity is reduced by 1 bandwidth-unit, the maximum data flow between a given source-destination node is also reduced by 1 bandwidth-unit. Therefore the goal of MIRA is accomplished by selecting paths that contain as few critical links as possible. However MIRA suffers from two weaknesses. First, MIRA is computationally expensive, because it needs to perform a lot of maximum data flow calculations, and each max flow computation is $O(N^3)$ [SWW01], where N is the number of nodes in the network. Compare to the runtime of SPF, which is $O(NM)$ (M is the number of links in the network), the runtime for MIRA is several magnitudes higher. Second, in some situations, MIRA may continuously choose the same set of links to route traffics on. To illustrate this point, consider the situation where there exist two distinct routes with identical residual bandwidth connecting the same source-destination pair (S, D). Initially all links in both route are classified as critical links. When a request associated with (S, D) arrives, given sufficient, one of the two routes will be chosen to serve this request. Afterwards, all the links in the chosen route are no longer critical, but all links in the other route remain critical links. This means subsequent requests will be routed on the same route as the first request until resources along this route are exhausted.

Some other related work includes profile-based routing and variations of OSPF routing heuristics proposed in [FT00] and [SWW01] respectively. These two routing algorithms require a traffic profile or an estimated demand matrix structure that approximates the bandwidth demands between each pair of source-destination nodes at different times. They will not be used for performance comparison purposes because they are not strictly online routing algorithms. Instead, we will compare SPF, MIRA and DORA in the performance evaluation section.

4 Dynamic Online Routing Algorithm

We consider the problem of setting up bandwidth guaranteed paths in a MPLS network. Each path setup request arrives one-by-one and we do not know the arrival time or size of future requests. The size of a request refers to the bandwidth requirement of the path to be setup. Each request demands a path with reserved bandwidth to be setup between an ingress (source) node and an egress (destination) node.

DORA is separated into two stages. Stage one is executed whenever a topology change has occurred, and stage two is executed whenever a path setup request arrives to the network. In the first stage, the key operation is to assign path potential value (PPV) to each link with respect to each source-destination pair. PPV reflects how likely a particular link will be part of some potential paths between some source-destination pairs in the network. A large PPV link value implies that this link will likely be part of many potential paths and thus routing over this link should be avoided whenever possible. A small PPV link value means that there are less potential paths using this link and therefore it is more desirable to use this link than others with larger PPV value. Each source-destination (S, D) is associated with an array, $PPV_{(S,D)}$, of size equal to the number of network links and each array element is initialized to zero. The way that $PPV_{(S,D)}$ array elements are calculated is based on which links are included in the disjointed set of paths for each source-destination pairs and it is described in detail in the pseudo-code listing for DORA. The main idea is to go through each source-destination pair (S, D), reduce one from $PPV_{(S,D)}(L)$ if the link L appears in the disjointed path set for (S, D), and add one to $PPV_{(S,D)}(L)$ each time L appears in any of the disjointed path sets associated to any other source-destination pairs. In the second stage, the PPV for each link is combined with the reciprocal of the current residual bandwidth of this link to form the link weight. The content of the link weight is controlled by a user parameter BWP (bandwidth proportion), which takes on values between 0.0 and 1.0. For example, if BWP equals 0.7, this implies that 70% of the link weight is contributed by the link's residual bandwidth and 30% of the link weight is contributed by the associated PPV.

Stage 1:

1. For each ingress-egress pair (S, D), determine the set of all disjointed paths $DP_{(S,D)}$. One possible way is to use Dijsktra's algorithm to find a shortest path (in terms of number hops) for (S, D), add this path to $DP_{(S,D)}$, and then remove all links that are part of the resulting path, and repeat these steps until D is no longer reachable from S.

2. For ingress-egress pair (S_1, D_1), construct the $PPV(S_1, D_1)$ array, and initialize all entries to zero. The size of the array is equal to the number of network links.

3. For each ingress-egress pair (S_1, D_1):

   a) Go through each link in the network and if a link L is part of any paths in $DP_{(S_1,D_1)}$, subtract 1 from $PPV_{(S_1,D_1)}(L)$.

   b) For all the ingress-egress pair other than (S_1, D_1), inspect each link L and determine the number of times, X, that L appears in $DP_{(S,D)}$ where (S, D) not equal to (S_1, D_1). Increment $PPV_{(S,D)}(L)$ by X.

4. Repeat step 3 - 4 for all the other ingress-egress pairs.

5. For each ingress-egress pair (S, D), normalize all entries in $PPV_{(S,D)}$, with the smallest PPV element over all ingress-egress pairs equal to 0 and the largest PPV element over all ingress-egress pairs equal to 100. Let $NPPV_{(S,D)}(L)$ to be equal to the normalized value of $PPV_{(S,D)}(L)$.

Stage 2:

1. Suppose a request arrives for path setup between (S_1, D_1) with Y amount of bandwidth. Remove links with residual bandwidth less than the requested bandwidth Y.

2. For each network link L, determine its residual bandwidth $RB(L)$, take the reciprocal of $RB(L)$ and normalize $RB(L) - 1$ to the range 0 to 100, with the smallest $RB(L) - 1$ equal to 0 and the largest $RB(L) - 1$ equal to 100. Let $NRB(L)$ to be equal to the normalized value of $RB(L) - 1$.

3. For the ingress-egress pair (S_1, D_1), construct a link weight table $LWT_{(S_1,D_1)}$, and using the following equation to obtain $LWT_{(S_1,D_1)}(L)$:

$$LWT_{(S_1,D_1)}(L) = NPPV_{(S,D)}(L) \times (1 - BWP) + NRB(L) \times BWP \quad (1)$$

4. Run Dijsktra's algorithm to compute a link weight-optimized path between (S_1, D_1).

5 Performance Evaluation

In this section, we will first describe the set of experiments used to evaluate the performance of DORA, and then we will comment on the results of the experiments. All experiment scenarios are simulated using ns-2 [BEF$^+$00]. We compare the performance of DORA with different BWP parameter values (0.1, 0.5, and 0.9) against that of SPF and MIRA. Two main performance metrics of interest are the path setup rejection ratio and the percentage of paths successfully rerouted upon topology change. The network topology used in the experiments

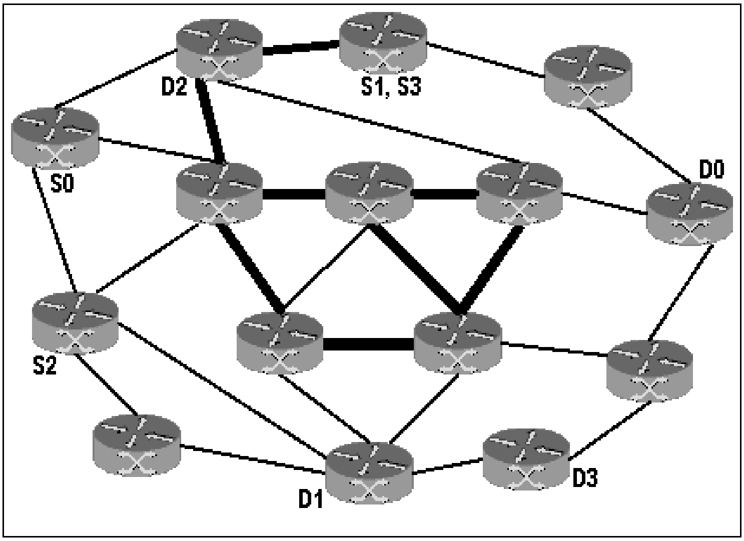


Fig. 1. Network topology used in the experiments. The thicker lines represent links with 48MBytes of reservable bandwidth while the thinner lines represent links with 12MBytes of reservable bandwidth.

represents a small ISP's backbone network and is shown in Figure 1. The figure also shows the location of 4 different source-destination pairs, identified by (S_0, D_0), (S_1, D_1), (S_2, D_2) and (S_3, D_3).

Three different experiment scenarios are considered. The first two experiments focus on the path rejection ratio, which indicates the percentage of path setups that are rejected due to insufficient resources. The last experiment studies the routing algorithm behaviors upon link failures. The size of each path setup request or the bandwidth requirement of the path is uniformly distributed among 10KB, 20KB, 30KB and 40KB. In experiment 1, a total of 2000 static path setup requests are sent to the network. Static paths resemble long-lived MPLS tunnels and once they are established, they will stay in the network forever. In experiment 2, we first load up the network with 200 static paths, and then we send 1800 dynamic path setup requests to the network. Dynamic paths represent short-lived MPLS tunnels. The arrival time of dynamic path setup requests at the network is based on a Poisson distribution with mean λ=40 requests per time-unit and each dynamic path has a holding time based on an Exponential distribution with mean μ=10 time-unit. The setup of experiment 3 is the same as that of experiment 1 with an additional setting - a randomly chosen link is taken down just before network resources are saturated. The number of paths requiring reroute and the percentage of successful reroutes are recorded after a link is taken down.

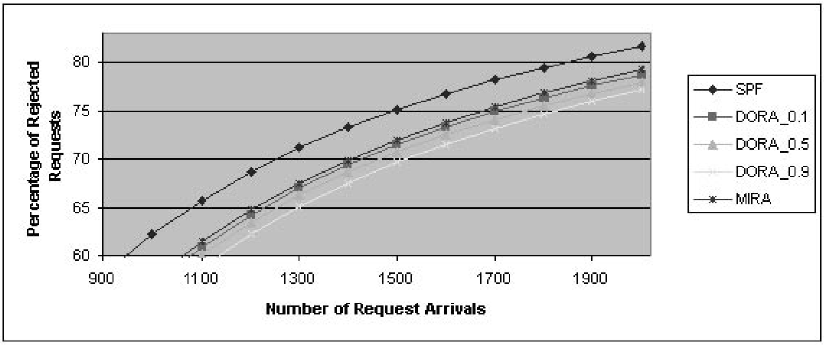


Fig. 2. Static Path Setup (Experiment 1): Percentage of Rejected Requests between Request #850 and #2000.

Figure 2 shows the partial result for experiment 1, which involves only static path setup requests. According to the figure, DORA_0.9 (DORA with $BWP = 0.9$) rejects the fewest number of requests, followed by DORA_0.5, DORA_0.1, MIRA and finally SPF. The result before setup request #850 is similar - that is the relative positions of the curves are the same during the course of the simulation. Since static paths remain in the network forever, after all network resources are exhausted, any incoming path request will be rejected. This can be observed by the fact that all curves in the figure approaches 100% as the number of request arrival increases.

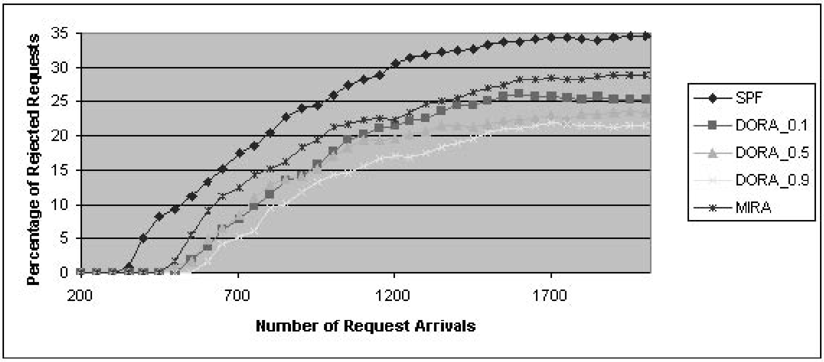


Fig. 3. Static-Dynamic Path Setup (Experiment 2): Percentage of Rejected Requests.

The result for experiment 2, which involves both static and dynamic path setup requests, is shown in Figure 3. In the figure, all curves grow irregularly until around request #1800, at which all curves enter steady state and stay relatively flat. Similar to the previous experiment, DORA_0.9 rejects the least percentage of requests, followed by DORA_0.5, DORA_0.1, MIRA and lastly SPF. During steady state, DORA_0.9 rejects about 26% less requests than MIRA, and DORA_0.1 shows 12% improvement on number of rejected requests over MIRA. The improvement over SPF is even more significant as DORA_0.9 and DORA_0.1 rejects around 37% and 27% less requests than SPF, respectively.

Experiment 3 is equivalent to experiment 1 except that at different points in time, a randomly chosen link is taken down and the number of paths requiring reroute and the percentage of successful reroutes are recorded. In experiment 1, the condition where all incoming requests are rejected due to insufficient resources occurs when just above 30% of the total network capacity has been occupied. We defined point A, B, and C to be the case where 20%, 25%, 30% of the total network capacity has been saturated. There are a total of 26 network links which yields a total of 390 experiment trials (e.g. 26 × 3 × 5). At each link failure point (A, B, and C), we compute the average number of paths requiring reroute, the standard deviations on the number of paths requiring reroute, and the percentage of successful reroutes. The results are shown in Figure 4, Figure 5, and Figure 6.

Figure 4 shows the average number of paths requiring reroute increases, as the network resource is closer to exhaustion. The least number of paths are required to be rerouted upon a link failure by using DORA_0.5, followed by DORA_0.1, DORA_0.9, MIRA and finally SPF. The standard deviation value for the number of paths requiring reroute is a direct indication of the algorithm's ability to spread path setups evenly across the network.

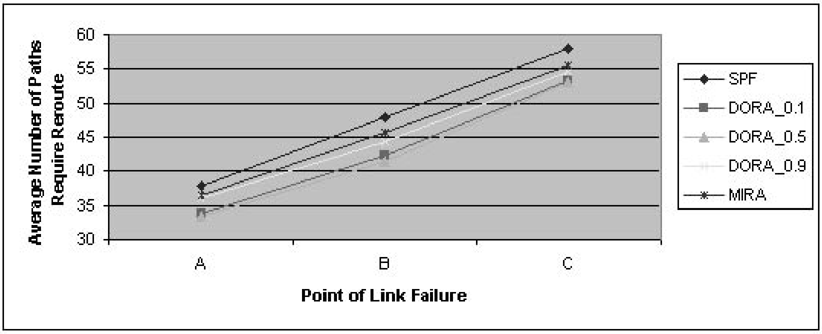


Fig. 4. Average Number of Paths Requiring Reroute at Different Failure Points.

Figure 5 shows that DORA_0.5 has the lowest standard deviation value, meaning that it is the most capable of spreading path setups across the network. In addition, both DORA_0.1 and DORA_0.9 have lower standard deviation value than either MIRA and SPF at all link failure points.

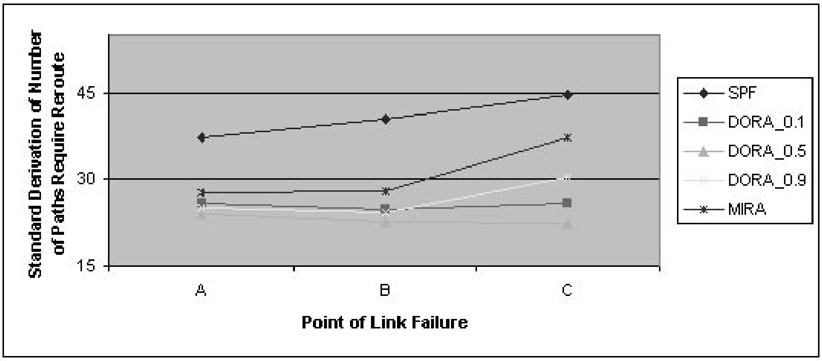


Fig. 5. Standard Deviation of Number of Paths Requiring Reroute at Different Failure Points.

Figure 6 shows the percentage of successful reroutes upon link failure. According to the figure, the curve for all algorithms declines, as the amount of network resource is closer to saturation. DORA_0.5 again performs the best among all algorithms with the highest successful reroutes percentage, followed by DORA_0.9, DORA_0.1, MIRA and lastly SPF. DORA_0.5 is able to obtain about 2%, 7.9%, and 6.5% more successful reroutes than MIRA at link failure points A, B, and C respectively. The improvement over SPF is more significant, as DORA_0.5 is able to obtain 18.28%, 25.53%, and 37.4% more successful reroutes at link failure points A, B, and C respectively. The results for experiment 4 suggest that a good mix of path potential value and residual bandwidth utilization yield the best performance in situations where link failure is commonplace.

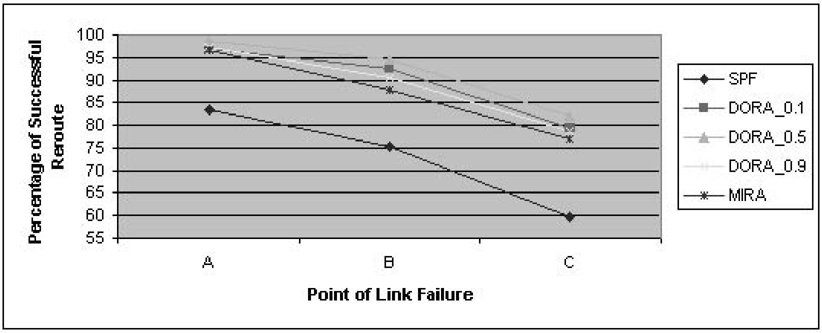


Fig. 6. Percentage of Successful Reroutes at Different Failure Points.

Next we will examine and compare the computation complexity of the routing algorithms used in the experiment. Consider a network consisting of N nodes and M links. Let D be the largest degree of any node and P be the number of source-destination pairs. Table 1 shows the computation complexity between the shortest-path algorithm (SPF), the minimum interference routing algorithm (MIRA), and our algorithm (DORA). SPF is least expensive in terms of runtime complexity, but it offers a much worse performance than other algorithms as shown in the previous experiments. The second stage of DORA is equally inexpensive as SPF and is executed during each request arrival. The first stage of DORA is performed only upon network topology change. In the absolute worst case scenario where $D = O(M)$ and $P = O(N^2)$, the runtime for stage one of DORA is several magnitudes higher than that of SPF, but still lower than that of MIRA.

Table 1. A comparison of computation complexity between different routing algorithms.

Algorithm	Computation Complexity
SPF	$O(MN)$
MIRA	$O(N^5) + O(M^2)$
DORA Stage 1	$O(N^3M^2)$
DORA Stage 2	$O(MN)$

6 Conclusion

In this paper, we have introduced Dynamic Online Routing Algorithm (DORA) for computing bandwidth guaranteed paths in MPLS networks. It combines the path potential value and current residual bandwidth to construct the link weight table with respect to each source-destination pair. A weight-optimized path based on the associated link weight table is then computed and returned by DORA. In the performance evaluation section, we have shown that DORA rejects fewer path setup requests than both SPF and MIRA. Furthermore, DORA attains a higher successful reroutes percentage upon link failures than both SPF and MIRA. In addition, the runtime complexity of DORA is less than that of MIRA, and it has an equivalent computation complexity to SPF when topology change is infrequent. When topology change does occur, it will trigger the execution of stage 1 in DORA. The cost of stage 1 operation could be reduced by using a better scheme for computing the set of disjointed paths with respect to each source-destination pair. Additionally, instead of recalculating the set of disjointed paths every time a topology change occurs, recompute only the affected disjointed paths.

One possible extension to DORA is to use past knowledge to estimate the future demand size for each source-destination pair. For instance, instead of incrementing and decrementing the PPV of a link by one, we may modify the PPV by a value higher than one to reflect a larger expected demand size for a given source-destination pair. Such knowledge could be inferred from constant network monitoring and measurements, or derived from the service level agreement between the customer and the service provider.

References

[AC00] G. Ahn and W. Chun. Design and implementation of mpls network simulator supporting ldp and cr-ldp. In *Proceedings of ICON*, 2000.

[AMA^{+}99] D. Awudche, J. Malcolm, J. Agogbua, M. O'Dell, and J. McManus. Requirements for traffic enginering over mpls. *RFC2702*, 1999.

[AMO93] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows: Theory, Algorithms, and Applications.* Prentice Hall, 1993.

[BEF^{+}00] L. Breslau, D. Estrin, K. Fall, S. Floyd, J. Heidemann, A. Helmy, P. Huang, S. McCanne, K. Varadhan, Y. Xu, and H. Yu. Advances in network simulation. *IEEE Computer Magazine*, 33(5):59–67, May 2000.

[Bla01] U. Black. *MPLS and Label Switching Networks.* Prentice Hall, 2001.

[FT00] B. Fortz and Mikkel Thorup. Internet traffic engineering by optimizing ospf weights. In *Proceedings of INFOCOM*, 2000.

[GJ79] M. Garey and D. Johnson. *Computer and Intractability.* W. H. Freeman, 1979.

[KL00] M. Kodialam and T.V. Lakshman. Minimum interference routing with applications to mpls traffic engineering. In *Proceedings of INFOCOM*, 2000.

[KYK02] D. Katz, D. Yeung, and K. Kompella. Traffic engineering extensions to ospf. *Internet Draft draft-katz-yeung-ospf-traffic-06.txt*, 2002.

[LS01] T. Li and H. Smit. Is-is extensions for traffic engineering. *Internet Draft draft-ietf-isis-traffic-04.txt*, 2001.

[RVC01] E. Rosen, A. Viswanathan, and R. Callon. Multiprotocol label switching architecture. *RFC3031*, 2001.

[SWW01] S. Suri, M. Waldvogel, and P. R. Warkhede. Profile-based routing: A new framework for mpls traffic engineering. In *Quality of Future Internet Services, Lecture Notes in Computer Science 2156.* Springer Verlag, Sept 2001.

[XHBN00] X. Xiao, A. Hanna, B. Bailey, and L. M. Ni. Traffic engineering with mpls in the internet. *IEEE Network Magazine*, pages 28–33, March 2000.

Optimal Capacity Provisioning for Label Switched Paths in MPLS Networks

C. Bruni[1], C. Scoglio[2], and S. Vergari[1]

[1] Dipartimento di Informatica e Sistemistica, University of Rome “La Sapienza”,
Via Eudossiana 18, 00184 Rome, Italy
{brunic, vergari}@dis.uniroma1.it

[2] Broadband and Wireless Networking Lab, School of Electrical and Computer
Engineering, Georgia Institute of Technology, Atlanta GA 30332, USA
caterina@ece.gatech.edu

Abstract. Optimal control is a possible approach to Internet traffic engineering, which aims to achieve QoS guarantees and efficiency in network resources use. The goal can be better achieved by using the Multi-Protocol Label Switching technique (MPLS), which provides increased scalability, manageability and enhanced QoS functions in IP-based networks. In this context, this paper proposes a method to find the optimal capacity provisioning for a Label Switched Path (LSP) of a MPLS network. The optimal capacity allocation for a given time interval is computed with respect to a quadratic cost function including a switching cost and a management cost for the whole network. The unique optimal solution is analytically computed assuming the knowledge of the offered traffic for the whole control interval. Furthermore, a sub-optimal on line solution is proposed which only requires the knowledge of a narrow sliding window of the offered traffic. Optimal and sub-optimal solutions are compared with respect to a simulated case study, enlightening the simplicity and, at the same time, the effectiveness of the second one.

1 Introduction

The growth of Internet has made evident the need for Traffic Engineering (TE) which became an essential tool for Internet Service Providers (ISPs) to optimize the network resources utilization in order to achieve QoS guarantees. Today, QoS-based services are offered in terms of contract agreements between the ISP and its customers.

Several architectures for supporting QoS have been developed[1,2]. A strategy proposed in IETF[3] is to consider two different types of services: *Corporate-service* and *Customer-service.*

Corporate-service is based on a contract, the Service Level Agreement (SLA), between the customer and the ISP, which is valid for a reasonable period of time and for a large amount of bandwidth. This type of agreement can be considered ”*quasi permanent SLA*” and it is particularly appropriate for companies who frequently require QoS guarantees and connections among distant network

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 947–958, 2002.

nodes (e.g. branch offices). The contract could also consider "*quasi on-demand*" requests, i.e., bandwidth requests which can be satisfied within a determinate period of time and not immediately "*on-demand*".

On the other hand, customer-service is not based on quasi permanent SLAs but supports the setup of dynamic QoS sessions.

From the above considerations, it turns out that TE can be exercised on two time-scales depending on the service nature:

Long-term: the traffic requests are predicted on a wide interval (days-weeks-months) by the existing long-term SLA contract;
Short-term: decisions are based on the observed state of the operational network on a short interval (minutes-hours).

Multi-Protocol Label Switching (MPLS) can be used to perform TE [4]. It has been shown that MPLS provides increased scalability, manageability and QoS functions to IP-based networks[5].

MPLS is the convergence of connection-oriented forwarding techniques and the Internet's routing protocols[6,7]. MPLS directs the flow of IP packets along a predetermined path inside the network, called Label Switched Path (LSP). The main concept of MPLS is to pad a label on each packet. Packets are assigned a short fixed length label that summarize the destination, the precedence, QoS information and route.

The LSP setup problem has been approached in order to reduce the number of LSPs in the network and to get an optimal resources utilization[8,9]. Another important issue is the MPLS network dimensioning: the objective is to accommodate all expected demands without overloading any part of the network.

In this paper we consider two routers connected by a direct LSP and we formulate the LSP capacity dimensioning problem as an optimization problem. We will assume that, at each time, the bandwidth request between the two considered routers is completely satisfied, partly by the LSP direct connection and partly by an alternative IP connection (this latter if the LSP capacity is not enough to satisfy the request). First we assume to know the bandwidth requests between the two routers over the whole control time interval [0, T]. This is a quite unlikely assumption because it is not possible to exactly predict the traffic profile that will be offered to the network, due to the nature of IP traffic. However, we observe some properties of the optimal solution that allow us to reduce the need for the bandwidth request knowledge just to a sliding window over [0, T], centered on the current time. In this case we obtain a sub-optimal, almost "on line", solution.

In the context of long-term TE and in particular of corporate-service agreements which consider "*quasi on-demand*" bandwidth requests, the knowledge of a narrow sliding window over [0,T] is a much more likely assumption because an ahead booking can be considered. Moreover, the proposed solution is completely independent of any stochastic assumption on the Internet traffic demand behaviour.

In the next Section, we formulate the optimal LSP capacity provisioning problem with the assumption of knowing the offered traffic over the whole control interval. Then the solution is found with respect to a quadratic cost function and subjected to a physical lower bound constraint. In Section 3, an on line sub-optimal solution of the same problem is proposed, obtained by reducing the hypothesis on the bandwidth requests knowledge. In the same Section, an analysis has been performed about the approximation offered by the sub-optimal solution with respect to the optimal one. In Section 4, some concluding remarks are given with respect to a simulated case study; the results confirmed the high approximation level of the sub-optimal solution, together with relevant advantages related to the reduction in the required information.

2 Optimal Solution for the Capacity Provisioning Problem

Let us denote by [0, T] the fixed control time interval over which we assume to know the bandwidth requests between two fixed routers. We consider an uniform discretization of [0, T] and denote by $k = 1, ..., N$, the discrete time variable. Furthermore, let $b(k)$ and $x(k)$ respectively denote the bandwidth request and the LSP capacity at time k. We assume that $b\,(k) \in [0, A]$, $k = 1, ..., N$, where A denotes the bandwidth availability on the LSP. We can consider the LSP capacity like a simple linear discrete time dynamical system:

$$x(k) = x(k-1) + \mathit{\Delta}(k), \qquad k = 1, 2, ..., N \tag{1}$$

where the initial state $x(0) = x_0$ is assumed known and positive and the control variable $\mathit{\Delta}(k)$ represents the capacity variation of the LSP at time k.

The LSP capacity is constrained by the following inequalities:

$$x(k) \geq 0, \qquad k = 1, 2, ..., N \tag{2}$$

which has an obvious physical meaning.

For the above LSP capacity provisioning problem, we have defined a cost function in order to take into account the most relevant cost terms and, in the meantime, to get a handling mathematical formulation. In particular we consider the following cost terms:

- *LSP cost:* it takes into account the cost due to the reserved capacity used to forward packets in MPLS mode. This cost, at time k, is assumed proportional to the LSP capacity:

  $$J_l(k) = c_l \cdot x(k) \tag{3}$$

  where $c_l > 0$ is the unitary cost for LSP capacity allocation.
- *Excess cost:* it takes mainly into account a cost due to packets switching performed in IP mode and their routing on an alternative path that occurs when the LSP capacity is less than the bandwidth request ($x(k) < b(k)$). Following the criteria that forwarding packets in MPLS mode is less expensive

than IP mode, we assume the unitary cost coefficient c_e for the bandwidth request not allocated on the LSP, greater than c_l. To emphasize the advantage of MPLS techniques and to promote their utilization, we consider the above mentioned cost depending quadratically on the difference between the LSP capacity and the bandwidth request. On the other hand, it may happen that, at a generic time k, the LSP capacity is greater than the bandwidth request$(x(k) > b(k))$. In this case a certain amount of bandwidth is reserved without utilization: we have assumed to penalize this event with a cost depending quadratically on the amount of waste bandwidth. For simplicity we consider the coefficient per unit of waste bandwidth equal again to c_e. From the above consideration, at time k, we have the following excess cost term:

$$J_e(k) = c_e \cdot [b(k) - x(k)]^2 \tag{4}$$

where, as already said, $c_e > c_l$.

- *Dimensioning variation cost:* it takes into account the LSP dimensioning variation cost. Each change of LSP capacity is charged in order to avoid too much wide LSP capacity re-dimensioning, which in turns affects the dimensioning of the other LSPs in the MPLS network. The same term can also take into account the so called *signalling cost* which occurs at each LSP capacity variation. The dimensioning variation cost at time k is assumed to depend quadratically on the size variation of LSP capacity:

$$J_v(k) = c_v \cdot \Delta^2(k) \tag{5}$$

where $c_v > 0$ is the unitary dimensioning variation cost of the LSP.

It follows that the total cost function in the control interval is:

$$J_t = \sum_{k=1}^{N} c_l x(k) + \sum_{k=1}^{N} c_e [b(k) - x(k)]^2 + \sum_{k=1}^{N} c_v \Delta^2(k) \quad . \tag{6}$$

From (1), the cost function (6) can be rewritten as follows:

$$J_t = c_e \sum_{k=1}^{N} b^2(k) + c_l \sum_{k=1}^{N} x(k) - 2c_e \sum_{k=1}^{N} b(k)x(k) + (c_v + c_e)x^2(N) +$$
$$+ c_v x_0^2 + (2c_v + c_e) \sum_{k=1}^{N-1} x^2(k) - 2c_v \sum_{k=1}^{N-1} x(k+1)x(k) - 2c_v x_0 x(1) \quad .$$

Our aim is to minimize the total cost J_t with respect to the variable $x(k)$, $k = 1, .., N$, in the presence of constraints (2). Let us note that the terms $c_e \cdot \sum_{k=1}^{N} b^2(k)$ and $c_v \cdot x_0^2$ do not depend on $x(k)$, $k = 1, .., N$, therefore we do not consider them in the cost minimization. We can, at this point, formulate the following quadratic programming problem.

Problem 1. *Find a global minimum for the cost function:*

$$J\left(x\right) = x^T \cdot H_N \cdot x + f^T \cdot x \tag{7}$$

in the admissible set:

$$D = \left\{ x \in \mathbb{R}^N : \quad x \geq 0 \right\} \tag{8}$$

where x and f are the following N-vectors:

$$x = \begin{pmatrix} x(1) \\ \vdots \\ x(N) \end{pmatrix}, \qquad f = \begin{pmatrix} c_l - 2c_e \cdot b(1) - 2c_v \cdot x_0 \\ c_l - 2c_e \cdot b(2) \\ \vdots \\ c_l - 2c_e \cdot b(N) \end{pmatrix} \tag{9}$$

and H_N is the following $N \times N$ matrix:

$$H_N = \begin{pmatrix} 2c_v + c_e & -c_v & 0 & . & 0 & 0 \\ -c_v & 2c_v + c_e & -c_v & . & 0 & 0 \\ 0 & -c_v & . & . & . & . \\ . & . & . & . & -c_v & 0 \\ 0 & 0 & . & -c_v & 2c_v + c_e & -c_v \\ 0 & 0 & . & 0 & -c_v & c_v + c_e \end{pmatrix} . \tag{10}$$

The matrix H_N is definite positive, as can be easily proved by exploiting some results in [10], so that J is strictly convex in $\mathbb{R}^N$.

The solution of Problem 1 is given in the following theorem.

Theorem 2. *Assuming $c_e > c_l$, $b(k) \geq \frac{1}{2}$, $k = 1, ..., N$, the unique solution of Problem 1 is:*

$$x^o = -\frac{1}{2} H_N^{-1} \cdot f \tag{11}$$

Proof. Taking the strict convexity of J into account, the unique global minimum of J in $\mathbb{R}^N$ is the solution of the equation:

$$\left(\frac{dJ}{dx}\right)^T_{x^o} = 2H_N \cdot x^o + f = 0$$

that is (11). In order to prove that (11) is also the unique solution of Problem 1, we will verify that $x^o \in D$, that is $x^o\left(k\right) \geq 0$, $k = 1, .., N$. The generic component of x^o is:

$$x^o\left(k\right) = -\frac{1}{2} \sum_{j=1}^{N} \left(H_N^{-1}\right)_{kj} \cdot f\left(j\right) \qquad k = 1, 2, ..., N \ . \tag{12}$$

By suitably handling a result given in [10] about the analytical expression of H_N^{-1}, we have:

$$\left(H_N^{-1}\right)_{ij} = \frac{c_v^{|i-j|}}{\det\left\{H_N\right\}} \det\left\{H_{N-\max\{i,\,j\}}\right\} \det\left\{K_{\min\{i,\,j\}-1}\right\} \tag{13}$$

where H_i, $i = 1, ..., N$, is an $i \times i$ matrix defined according to (10) and K_i is the following $i \times i$ matrix:

$$K_i = \begin{pmatrix} 2c_v + c_e & -c_v & 0 & . & 0 \\ -c_v & 2c_v + c_e & . & . & . \\ 0 & . & . & -c_v & 0 \\ . & . & -c_v & 2c_v + c_e & -c_v \\ 0 & . & 0 & -c_v & 2c_v + c_e \end{pmatrix}, \qquad i = 1, 2, ..., N \ .$$

Noting that H_i and K_i, $i = 1, ..., N$, are definite positive matrices, as can be proved by exploiting again results in [10], the positivity of $\left(H_N^{-1}\right)_{kj}$ follows from (13) for $k, j = 1, ..., N$. The positivity of $x^o(k)$, $k = 1, ..., N$, is then implied by the positivity of $-f(j)$, $j = 1, ..., N$. This, in turn, is an obvious consequence of the assumptions and of the positivity of x_0. ■

Remark 3. It is worth noting that the optimal solution $x^o(k)$, for each k, depends on all the samples $b(j)$, $j = 1, ..., N$, as it clearly results from (12).

3 Sub-optimal on Line Solution

Although the hypothesis of complete knowledge of Internet traffic demand on the control discrete time interval [0, N] is partially supported by long-term TE framework, our aim, in this Section, is to reduce this hypothesis. Indeed, we will show that, for the particular structure of the inverse matrix H_N^{-1}, we can motivate a sub-optimal solution assuming to know just a narrow sliding window on the bandwidth profile, centered at the current time, much smaller than the total time interval [0, N] considered before. As a consequence, the structure which characterizes the sub-optimal solution can be implemented "*on line*", while the optimal one is clearly "off line".

In order to analyze the behaviour of the suboptimal solution we are going to introduce, let us define the following parameter:

$$\alpha_h = \max_{i,j\colon\, |i-j|=h} \left\{\left(H_N^{-1}\right)_{ij}\right\} \qquad h = 1, 2, ..., (N-1) \ . \tag{14}$$

The behaviour of the above parameter α_h has been numerically investigated for different values of c_v, c_e, c_l and N. The analysis has pointed out a monotone decreasing behaviour of α_h, as shown for instance in Fig.s 1, 2, 3.

Let us now give the definition of the sub-optimal solution for the Problem 1.

Definition 4. *For a fixed integer $N > 1$, let be $M \leq 2N - 3$ a positive odd integer.* We define the following sub-optimal solution:

$$x^{so} = -\frac{1}{2} P_{NM} \cdot f \tag{15}$$

where P_{NM} is the M-diagonal matrix of dimension $N \times N$ with entries:

$$(P_{NM})_{ij} = \begin{cases} \left(H_N^{-1}\right)_{ij} & for \quad |i-j| = 0, 1, ..., \frac{M-1}{2} \\ 0 \quad for & |i-j| = \frac{M+1}{2}, ..., (N-1) \end{cases} \tag{16}$$

Remark 5. From (16) it is obvious that the generic component $x^{so}(k)$, k=1,...,N, depends on a sliding window of no more than M components $f(j)$ of f. This means that the suboptimal solution, at each time k, requires the knowledge of bandwidth requests on a sub-interval containing no more than $\frac{M-1}{2}$ future samples $b(j)$.

In order to verify that x^{so} is a good approximation of x^o, we introduce an upper bound on the error, which depends on M and is sufficiently small when M is suitably chosen. In fact, considering the norm $\|\cdot\|_\infty$ and recalling that $b(k) \leq A$, $k = 1, ..., N$, from (9) it results:

$$\|f\|_\infty = \max_{k=1,...,N} \{|f(k)|\} \leq c_l + 2c_e A + 2c_v x_0 = 2C \quad .$$

Therefore

$$\begin{aligned} \|x^o - x^{so}\|_\infty = \max_{k=1,..,N} \{|x^o(k) - x^{so}(k)|\} &= \frac{1}{2} \left\| \left(H_N^{-1} - P_{NM}\right) f \right\|_\infty \leq \\ &\leq \frac{1}{2} \left\| \left(H_N^{-1} - P_{NM}\right) \right\|_\infty \cdot \|f\|_\infty \leq \\ &\leq C \max_{i,j=1,...,N} \left\{ \left| \left(H_N^{-1}\right)_{ij} - (P_{NM})_{ij} \right| \right\} \quad . \end{aligned}$$

From (13) and the definite positivity of H_i, K_i, we have $\left(H_N^{-1}\right)_{ij} > 0$, $i, j = 1, ..., N$. Then, from (16), taking the definition (14) of α_h into account together with its monotonic property, it results:

$$\begin{aligned} \max_{i,j=1,...N} \left\{ \left| \left(H_N^{-1}\right)_{ij} - (P_{NM})_{ij} \right| \right\} &= \max_{i,j:\ |i-j| = \frac{M+1}{2}, ..., (N-1)} \left\{ \left(H_N^{-1}\right)_{ij} \right\} = \\ &= \max_{h = \frac{M+1}{2}, ..., (N-1)} \alpha_h = \alpha_{\frac{M+1}{2}} \quad . \end{aligned}$$

Therefore we have:

$$\|x^o - x^{so}\|_\infty \leq C \alpha_{\frac{M+1}{2}} \quad . \tag{17}$$

In order to analyze the approximation level given by (17), it is useful to observe that if we set:

$$\frac{M+1}{2} = h \tag{18}$$

when M is an odd integer running from 1 to $(2N-3)$, h assumes the values 1, 2,...,$(N-1)$. Therefore we can rewrite (17) as follows:

$$\|x^o - x^{so}\|_\infty \leq C\alpha_h \quad .$$

and analyze the approximation level by exploiting the behaviour of α_h.

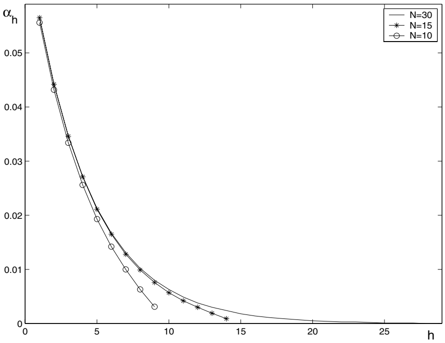


Fig. 1. Behaviour of α_h (c_v=50,c_e=3,c_l=1)

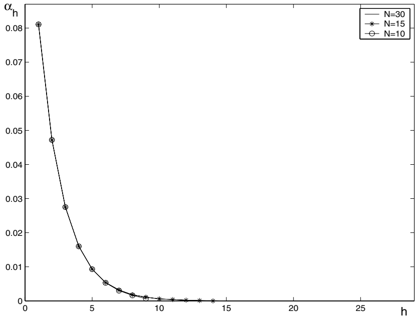


Fig. 2. Behaviour of α_h (c_v=10, c_e=3, c_l=1)

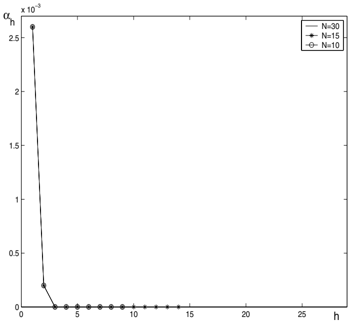


Fig. 3. Behaviour of α_h (c_v=3, c_e=30, c_l=1)

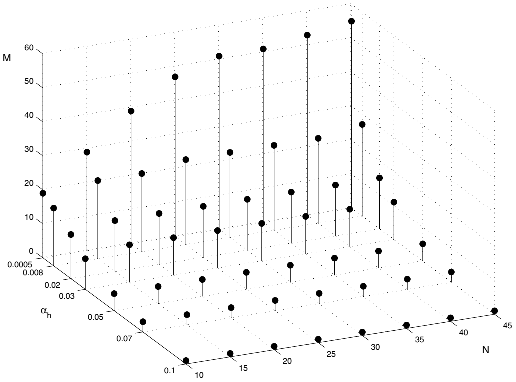


Fig. 4. Behaviour of M (c_v=50, c_e=3, c_l=1)

Remark 6. As it appears from Fig.s 1, 2, 3, α_h approaches quickly zero, for each fixed N, when h reaches a few unit value. Therefore, also when N increases, the approximation error can be kept low by assuming a suitable bounded value for h. For instance, in Fig. 1 and in Fig. 2 we have $\alpha_h < 10^{-2}$ when $h \simeq 7$ (which amounts to the knowledge of only six future samples of $b\,(j)$) and this virtually for every N greater than about ten. From Fig. 3, we observe that $\alpha_h < 3 \cdot 10^{-3}$, for every N, also if h is only equal to one (this means that $x^{so}\,(k)$, $k = 1, ..., N$, depends only on the current request $b(k)$ and no knowledge of the future is required). The same conclusion is also evidenced by Fig. 4 where the behaviour of the parameter M, numerically obtained by (14) taking (18) into account, is also given in a 3-dimensional representation for different values of α_h and N, assuming for instance $c_v = 50$, $c_e = 3$, $c_l = 1$. For α_h and N fixed, Fig. 4 allows to deduce the corresponding value for the parameter M. It appears that M quickly reaches a steady state value when N increases, for each fixed α_h.

4 An Application to Simulated Data

In order to test the application of the optimal and sub-optimal LSP capacity allocation procedures, we have considered a case study obtained by simulating a sequence of bandwidth requests.

To generate this bandwidth profile, we consider each request arrival time and each request death time as an event. Besides, we assume that two events occur at the same time with probability zero. In particular we simulate three stochastic processes:

- the first one generates the requests arrival times and it is simulated as a Poisson process with parameter $\lambda = \frac{1}{2}$;
- the second concerns the time duration of each request, and is characterized by an exponential distribution with parameter $\mu = \frac{1}{5}$;
- the last one is related to the amount of bandwidth of each request, and follows a uniform distribution on the integers of the interval [1, 10].

On the generated bandwidth profile we select a time window containing $N = 40$ samples. As initial state we consider $x_0 = 11.3$ corresponding to the average value of the bandwidth requests. Using the above data, we compute the optimal and the sub-optimal solutions considering, for the parameters c_v, c_e, c_l, the same values as in Fig.s 1, 2, 3. We have considered a sub-optimal solution based, for instance, on the knowledge of only 8 future samples, which means $M = 17$. Note that, for the choice $c_v = 50$, $c_e = 3$, $c_l = 1$ and assuming $A = 35$, Fig. 4 allows to guarantee an "a priori" approximation error with respect to the optimal solution not greater than 6.5.

In Fig.s 5, 6, 7 computed optimal and sub-optimal solutions together with the simulated bandwidth profile are shown. In particular, from Fig. 5 we easily verify the above expected approximation level.

Concerning the effects of the cost parameters on the optimal solution, we have the following remarks:

- noting that the optimal solution is defined modulo a positive factor in the cost function, we have normalized the cost coefficients assuming always $c_l = 1$;
- the parameter c_v, which weights the variation size cost, affects the behaviour of the optimal solution considerably; in particular the higher is the value of c_v, the smoother the solution becomes;

- the parameter c_e influences the fitting capability of the optimal solution with respect to the requested bandwidth reference. Note also that the same parameter influences the fitting capability of the sub-optimal solution with respect to the optimal one: this is due to the fact that, as $\frac{c_e}{c_v}$ increases, the matrix H_N approaches the identity matrix and consequently x^{so} tends to coincide with x^o.

A comparison between the optimal and the sub-optimal solution can be carried out both with reference to the instantaneous approximation error and to the related costs. For the first point, we observe that the maximum deviation between x^o and x^{so} is of about 3, 0.2, 0 respectively in the three considered cases. It is worth noting that virtually the same numerical results can be foreseen by exploiting the upper bound given by (17).

Concerning the related costs, considering for instance the first choice of parameters ($c_v = 50$, $c_e = 3$, $c_l = 1$), we obtain $J(x^o) = 6672$, while for the sub-optimal solution (with $M = 17$), we have $J(x^{so}) = 7141$, with an increase of about 6.56%. It appears that the cost increase is very low, when compared with the advantage (in the better case) of discarding 31 future samples $b(j)$. Finally note that, if we want to furthermore reduce the cost increase, we can increase M; assuming for instance $M = 21$ (10 future samples), the value of J for the corresponding sub-optimal solution, becomes 6827, which amounts to an increase of only 2.27%.

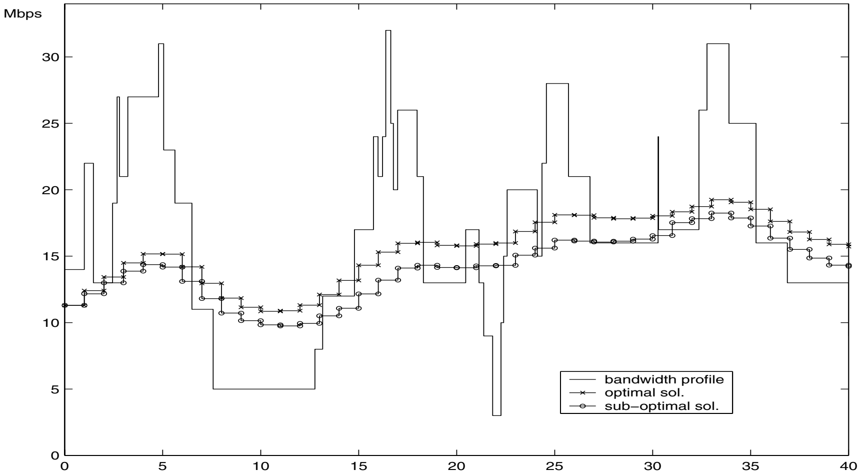


Fig. 5. Optimal and Sub-optimal solution (c_v=50, c_e=3, c_l=1, M=17)

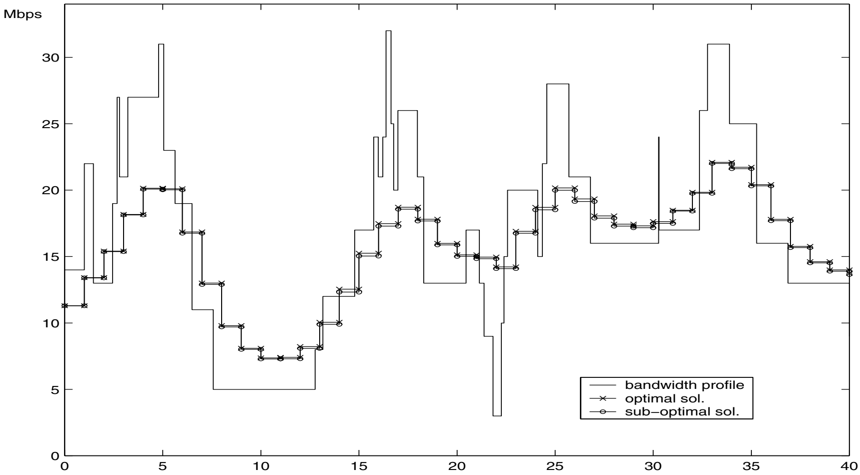


Fig. 6. Optimal and Sub-optimal solution (c_v=10, c_e=3, c_l=1, M=17)

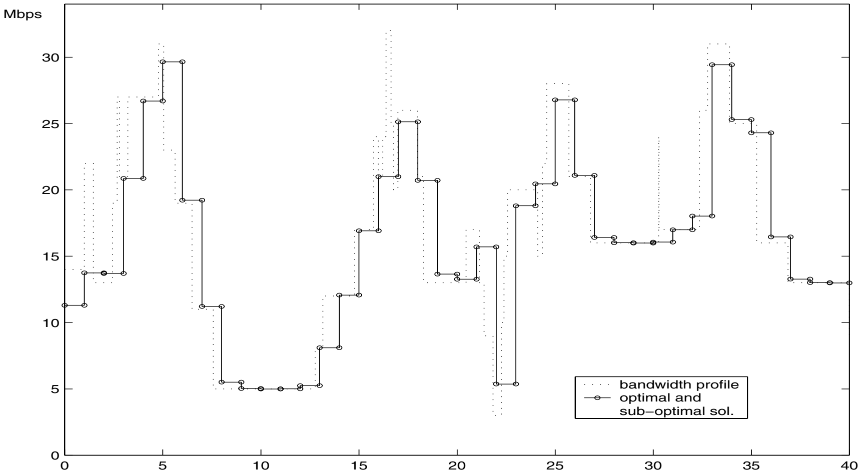


Fig. 7. Optimal and Sub-optimal solution (c_v=3, c_e=30, c_l=1, M=17)

5 Concluding Remarks

This paper provides a formal description of the optimal capacity provisioning problem for a label switched path in a MPLS network. In particular, by a suitable choice of the cost function, the above problem was reduced to a quadratic programming problem, whose closed form solution have been easily obtained.

This optimal solution requests the knowledge of all the future traffic in the control time interval. Being aware that future traffic knowledge is a quite unlikely assumption, by exploiting some properties of the optimal solution, we propose a sub-optimal one which offers the advantage of requiring the knowledge of future traffic only on a small sliding window over the control time interval and, at the same time, it offers a very good approximation level with respect to the optimal solution together with a very small increase of the cost.

Acknowledgments. The Authors are indebted to Professor I. F. Akyildiz for his encouragement and many valuable suggestions. Special thanks are also due to Professor F. Delli Priscoli for some useful discussions and to T. Anjali and J. C. Oliveira for their help in the realization of this paper.

References

[1] P. Trimintzios, D. Griffin, P. Georgatsos, D.Goderis, L.Georgiadis, C. Jacquenet, R. Egan: A Management and Control Architecture for Providing IP Differentiated Services in MPLS-Based Network. IEEE Communications Magazine, vol 39, n. 5, May 2001.

[2] R. Callon, E. Rosen, A. Viswanathan: MultiProtocol Label Switching Architecture. IETF, RFC 3031, January 2001.

[3] A. Bergsten, K. Nemeth, I. Cselenyi, G. Feher: Fundamental Questions Regarding End-to-End QoS. IETF, Internet Draft, July 2001.

[4] D. O. Awduche, J. Malcolm, J. Agogbua, M. O'Dell, J. McManus: Requirement for Traffic Engineering over MPLS. IETF, RFC 2702, September 1999.

[5] F. Gonzales, C. Chang, L. Chen, C. Lin: Using MultiProtocol Label Switching (MPLS) to Improve IP Network Traffic Engineering. Proc. Interdisciplinary Telecommunications Program, Spring 2000.

[6] G. J. Armitage: MPLS: the Magic Behind the Myths. IEEE Communications Magazine, vol 38, n. 1, Jan 2000.

[7] D. O. Awduche, A. Chiu, A. Elwalid, I. Widyaya, X. Xiao: A framework for Internet Traffic Engineering. IETF, Internet Draft, July 2001.

[8] C. Scoglio, T. Anjali, J. C. Oliveira, I. F. Akyildiz: A new Optimal Policy for Label Switched Path Setup in MPLS Network. Proc. 17th International Teletraffic Congress, Brazil, September 2001.

[9] H. Saito, Y. Miyao, M. Yoshida: Traffic Engineering Using Multiple Point-to-Point LSPs. Proc. INFOCOM 2000 (19th joint Conference of the IEEE Computer and Communication Societies), Tel Aviv, March 2000.

[10] C. F. Fischer, R. A. Usmani: Properties of some tridiagonal Matrices and their application to Boundary Value Problems. SIAM Journal on Numerical Analysis, Vol 6,n. 1, March 1969.

A New Class of Online Minimum-Interference Routing Algorithms

Ilias Iliadis and Daniel Bauer

IBM Research, Zurich Research Laboratory, 8803 Rüschlikon, Switzerland
{ili,dnb}@zurich.ibm.com

Abstract. On-line algorithms are essential for service providers to quickly set up bandwidth-guaranteed paths in their backbone or transport networks. A minimum-interference routing algorithm uses the information regarding the ingress–egress node pairs for selecting a path in the case of on-line connection requests. According to the notion of minimum interference, the path selected should have a minimum interference with paths considered to be critical for satisfying future requests. Here we introduce a new class of minimum-interference routing algorithms, called "simple minimum-interference routing algorithms" (SMIRA), that employ an efficient procedure. These algorithms use static network information comprising the topology and the information about ingress–egress node pairs, as well as the link residual bandwidth. Two typical algorithms belonging to this class are introduced, and their performance is evaluated by means of simulation. The numerical results obtained illustrate their efficiency, expressed in terms of throughput, and fairness.

1 Introduction

This paper deals with the issue of dynamic bandwidth provisioning in a network. This problem arises in several instances, such as in the context of dynamic label-switched path (LSP) setup in multiprotocol label switching (MPLS) [1] networks and in the context of routing virtual circuit requests over an ATM backbone network [2]. In particular, this paper considers the issue of establishing bandwidth-guaranteed connections in a network, in which connection-setup requests arrive one by one and future demands are unknown. This is referred to as an *on-line* algorithm, in contrast to an *off-line* algorithm that assumes *a priori* knowledge of the entire request sequence, including future requests. On-line algorithms are essential owing to the need of service providers to quickly set up bandwidth-guaranteed paths in their backbone or transport networks.

The primary routing problem consists of determining a path through the backbone network that a connection should follow. Clearly, the available bandwidth of all links on the path should be greater or equal to the requested bandwidth. If there is insufficient capacity, some of the connections cannot be established, and therefore are rejected. A significant body of work exists for the on-line path selection problem. Several path selection algorithms proposed in

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 959–971, 2002.

the literature aim at limiting the resource consumption so that network utilization is increased. The most important of these algorithms can be found in [3, 4]. The basic algorithms considered here along with a short description of their functionality are listed below. **Shortest-path routing** (SP) algorithms select a path with the least amount of aggregated cost. **Minimum-hop routing** algorithms select a path with the least number of links as that path uses the smallest amount of resources. When all links have the same cost, it is a special case of an SP algorithm. **Widest-shortest-path routing** (WSP) algorithms select a shortest path with the largest available (residual) bandwidth. **Shortest-widest-path routing** (SWP) algorithms select a widest path with the least amount of aggregated cost. A widest path is a path with maximum bottleneck bandwidth or, equivalently, a path with the largest In addition to these algorithms, a more sophisticated algorithm, called **minimum interference routing algorithm (MIRA)**, that uses the information regarding the ingress–egress pairs was recently developed [5]. Despite the fact that this algorithm uses this information pertaining to the past, present, and future requests, it is considered to be an online algorithm because the future connection requests are unknown. However, the effect of future requests is indirectly incorporated through the notion of the *minimum-interference routing.*

The second problem related to the primary routing is that of admission control. Routing algorithms can be categorized into two classes according to the control of admitting connections [6]. *Greedy* algorithms always establish a connection as long as sufficient capacity is available. *Trunk-reservation* algorithms reject a connection if assigning any of the existing paths could result in inefficient use of the remaining capacity regarding future connection requests [7].

Section 2 reviews the concept of minimum-interference routing and briefly describes the MIRA algorithm, the first such algorithm presented in [5]. The main contributions of this paper are presented in Sections 3 and 4. In Section 3, we present a new class of minimum-interference routing algorithms, called "simple minimum-interference routing algorithms" (SMIRA). This class of algorithms is not based on the principle of calculating maximum flows, but rather uses a more efficient (in terms of computational complexity) approach, hence the term "simple". In Section 4, we examine the efficiency of the algorithms proposed by means of simulation, and compare them with the SP, SWP, WSP, and MIRA algorithms. The efficiency assessment is based on performance metrics including the throughput, expressed as the total bandwidth of the routed (accepted) connections, the blocking-free range, and the fairness achieved among different ingress–egress node pairs. In particular, we demonstrate that the effectiveness of a given algorithm strongly depends on the performance criterion chosen. Finally, we draw conclusions in Section 5.

2 Minimum-Interference Routing

In this section, the notion of minimum-interference routing and the first MIRA algorithm are reviewed. Although for on-line algorithms future connection requests are unknown, the effect of future requests can be indirectly incorporated through the notion of *minimum-interference routing.* A new connection should follow a path that does not "interfere too much" with a path that may be critical to satisfy a future request. Note that this notion can only be used in conjunction with knowledge of all ingress–egress pairs. An explicit path between a given ingress–egress pair can in principle be calculated according to a defined interference criterion. For example, the criterion could be the maximization of the smallest maxflow value of all remaining ingress–egress pairs, the maximization of the weighted sum of the remaining maxflows (referred to as WSUM-MAX) [5], or the maximization of the maximum throughput of the corresponding multicommodity flow problem [7]. These problems are quite complex, therefore alternative, simplified approaches are highly desirable. One possibility is to turn the original problem into an equivalent shortest-path routing problem with appropriately selected link-cost metrics. In [5], this transformation is done as follows. The amount of interference on a particular ingress–egress pair, say (s, d), due to the routing of a connection between some other ingress–egress pair is defined as the decrease of the *maximum flow* value between (s, d). Then, the notion of *critical links* is introduced. These are links with the property that whenever a connection is routed over them, the maxflow values of one or more ingress–egress pairs decrease. For this definition, it turns out that the set of critical links corresponding to the (s, d) pair coincides with the links of all *minimum cuts* for this pair. Links are subsequently assigned weights that are an increasing function of their "criticality": the weight of a link is chosen to be the rate of change in the optimum solution of the original WSUM-MAX problem with respect to changing the residual capacity of the link. This choice results in critical links being assigned the highest weights. Finally, the actual explicit path is calculated using an SP algorithm.

3 Simple Minimum-Interference Routing Algorithms

In this section, we introduce the class of the simple minimum-interference routing algorithms (SMIRA). The term "simple" reflects the fact that they do not define the critical links according to maximum-flow calculations, as done by MIRA, but rather employ a simpler approach, which, as will be seen below, has a lower computational complexity than the MIRA maximum-flow approach.

To devise an alternative notion for the critical links we resort to the fundamental objective of minimum-interference routing, namely that a new connection must follow a path that interferes as little as possible with a path that may be critical to satisfy a future request. This requires that paths associated with future requests be taken into account and also that links associated with such critical paths be identified and weighted accordingly. One possible way to achieve this is the following. Let P be the set of ingress–egress node pairs, and suppose that the connection request is between nodes a and b. For every of the remaining ingress–egress pairs (s, d) we identify a set of critical paths, and each link of such a path is weighted accordingly. When this process has been completed, links having minimum weight are associated with paths that do not interfere with future requests. Routing the current connection request on a shortest path with respect to these weights results in a residual network in which the interference of the remaining ingress–egress pairs is kept to a minimum.

3.1 Critical Paths

There are several ways to identify a set of critical paths corresponding to the ingress–egress pair (s, d). Here we introduce a procedure for obtaining the set of critical paths called *K-widest-shortest-path under bottleneck elimination.* This procedure identifies a set of critical paths by making use of a WSP algorithm. The paths are enumerated in descending order of their significance. The algorithm starts with selecting the widest-shortest path between pair (s, d). Let $L_{sd}^{(1)}$ denote the set of links constituting this widest-shortest path, and let $btl_{sd}^{(1)}$ be the corresponding (bottleneck) bandwidth of this path. Let also $Btl_{sd}^{(1)}$ denote the subset of link(s) of this path whose residual bandwidth is equal to the bottleneck value $btl_{sd}^{(1)}$. The next (second) member is found by computing the shortest-widest path when the links of the set $Btl_{sd}^{(1)}$ are removed from the network. This procedure is repeated until either K paths are found or no more paths are available, whichever occurs first. It is realized by using the Dijkstra algorithm [8] in each iteration, therefore its complexity is of order $O(K(n \log n + m))$, n and m being the number of nodes and links, respectively.

An alternative procedure can be derived by using an SWP algorithm – or any other path-computation method – to enumerate the critical paths. This procedure, called *K-shortest-widest-path under bottleneck elimination*, is realized by using the Dijkstra algorithm twice in each iteration [3], therefore its complexity is also of order $O(K(n \log n + m))$.

For networks of practical relevance, it turns out that the value of K is typically a small number. Therefore, the complexity of the above procedures is of order $O(n \log n + m)$. On the other hand the complexity of the procedure for determining the set of critical links in the case of the MIRA algorithm is of order $O(n^2 \sqrt{m} + m^2)$ [5]. This complexity results from the two phases used by MIRA. The first consists of a maximum flow calculation with a computational complexity of $O(n^2 \sqrt{m})$. The second consists of the process of enumerating all links belonging to minimum cuts with a complexity of $O(m^2)$. Thus, the total complexity is of order $O(n^2 \sqrt{m} + m^2)$. In particular, in the case of sparse topologies, MIRA's complexity is of order $O(n^{2+\frac{1}{2}})$ as m is of order $O(n)$, whereas that of our algorithm is of order $O(n \log n)$. In the case of dense topologies, MIRA's complexity is of order $O(n^4)$ as m is of order $O(n^2)$, whereas that of our algorithm is of order $O(n^2)$. Therefore, our proposed procedures have a shorter expected execution time, justifying the use of the term "simple".

3.2 Link-Weight Assignment

Each link is initially assigned a static cost. For critical links this cost is scaled by a factor that includes two weight components. The first one is associated with the path(s) to which this link belongs. Naturally, higher weights are assigned to paths with higher significance. The second reflects the importance of the links constituting each of the critical paths.

We now turn our attention to the first weight component. Let $L_{sd}^{(i)}$ denote the set of the links constituting the i-th path associated with the ingress–egress pair (s, d), and $w_{sd}^{(i,l)}$ denote the corresponding weight contributed to link l of this set. Let also $btl_{sd}^{(i)}$ be the corresponding bandwidth of this path, and $Btl_{sd}^{(i)}$ be the subset of the corresponding bottleneck link(s). The paths are enumerated in descending order of their significance. In accordance, the links of the $L_{sd}^{(i)}$ path are weighted with a factor $v_{sd}^{(i)}$, which is a decreasing function in i such that the weight for link l should be proportional to this factor, i.e. $w_{sd}^{(i,l)} \sim v_{sd}^{(i)}$. Our rationale is the following: if all candidate paths for the new connection contain links that have already been marked by this process, i.e. the interference cannot be avoided, then the links associated with the most critical paths corresponding to the other ingress–egress pairs should be avoided by assigning them the highest weights. In this way the interference is relegated to secondary paths. There are infinitely many discounting value functions one could choose. Here we consider the following two functions: $v_{sd}^{(i)} = 1$ and $v_{sd}^{(i)} = (K - i + 1)/K$.

The next consideration is the weight assignment for the links constituting path $L_{sd}^{(i)}$. Let $g_{sd}^{(i,l)}$ denote the corresponding weight for link l. Intuition dictates that bottleneck links should be assigned a higher value than other links. Here again, there are infinitely many discounting value functions one could choose. In this paper, we consider two functions defined as follows. The inversely proportional function $g_{sd}^{(i,l)} = btl_{sd}^{(i)}/r(l)$ and the step function $g_{sd}^{(i,l)} = \lfloor btl_{sd}^{(i)}/r(l) \rfloor$, where $r(l)$ denotes the residual bandwidth of link $l \in L_{sd}^{(i)}$. Note that $btl_{sd}^{(i)}/r(l)$ is a decreasing function in $r(l)$ with $btl_{sd}^{(i)}/r(l) \leq 1$. Consequently, $\lfloor btl_{sd}^{(i)}/r(l) \rfloor = 1$, if and only if $r(l) = btl_{sd}^{(i)}$, otherwise $\lfloor btl_{sd}^{(i)}/r(l) \rfloor = 0$. Thus, in the case of the step function, it holds that $g_{sd}^{(i,l)} = \begin{cases} 1 & l \text{ is a bottleneck link of the path } L_{sd}^{(i)} \, , \\ 0 & \text{otherwise} \, . \end{cases}$

We proceed by choosing the weight contributed to link l to be proportional to the factors defined above, i.e. $w_{sd}^{(i,l)} \sim v_{sd}^{(i)}$ and $w_{sd}^{(i,l)} \sim g_{sd}^{(i,l)}$. In particular, we choose $w_{sd}^{(i,l)} = w \, v_{sd}^{(i)} \, g_{sd}^{(i,l)} \; , \quad (l \in L_{sd}^{(i)})$, where w is a scaling factor.

By taking into account all contributions, the weight of each link is then calculated by $w(l) = c(l) \left\{ 1 + \sum_{(s,d) \in P \setminus (a,b)} a_{sd} \, \sum_{i=1}^{K} \sum_{l \in L_{sd}^{(i)}} w_{sd}^{(i,l)} \right\}$, or

$$w(l) = c(l) \left\{ 1 + w \sum_{(s,d) \in P \setminus (a,b)} a_{sd} \sum_{i=1}^{K} v_{sd}^{(i)} \sum_{l \in L_{sd}^{(i)}} g_{sd}^{(i,l)} \right\} , \tag{1}$$

where $c(l)$ is a static cost that could, for example, depend on the capacity of the link, and a_{sd} is the weight of the ingress–egress pair (s, d). Note that for noncritical links it holds that $w(l) = c(l)$.

3.3 The SMIRA Algorithm

In summary the SMIRA algorithm is as follows:
INPUT: A graph $G(N, L)$, the residual bandwidth $r(l)$ for each link, and a set P of ingress–egress node pairs. An ingress node a and an egress node b between which a flow of D units have to be routed.
OUTPUT: A route between a and b with a bandwidth of D units, if it exists.
ALGORITHM:
1. Compute the K-critical paths $\forall (s, d) \in P \backslash (a, b)$. Let $L_{sd}^{(i)}$ be the set of the links constituting the i-th path.
2. Assign weight on each link according to Eq. (1).
3. Eliminate all links that have residual bandwidth smaller than D and form a reduced network.
4. Use Dijkstra's [8] algorithm to compute the shortest path in the reduced network with $w(l)$ as the weight of link l.
5. Route the demand of D units from a to b along this shortest path, and update the residual capacities.

SMIRA, as defined above, clearly constitutes a general class of algorithms containing an unlimited number of particular instances (implementations). Here we consider two particular algorithmic instances, and investigate their performance. The first is called *minimum-interference bottleneck-link-avoidance* algorithm (MI-BLA), and is derived from the following choices:
– The set of critical paths is obtained using the K-widest-shortest-path under bottleneck-elimination procedure, with K equal to 6.
– All paths are considered to have the same weight, i.e. $v_{sd}^{(i)} = 1$.
– Links are valued according to the step function, i.e. $g_{sd}^{(i,l)} = \left\lfloor btl_{sd}^{(i)}/r(l) \right\rfloor$.
– Setting $a_{sd} = 1$ and $w = 2$, the weight of each link is then calculated by

$$w(l) = c(l) \left\{ 1 + 2 \sum_{(s,d) \in P \backslash (a,b)} \sum_{i=1}^{K} \sum_{l \in L_{sd}^{(i)}} \left\lfloor \frac{btl_{sd}^{(i)}}{r(l)} \right\rfloor \right\} .$$

The second is called *minimum-interference path avoidance* algorithm (MI-PA), and is derived from the following choices:
– The set of critical paths is obtained using the K-widest-shortest-path under bottleneck-elimination procedure, with K equal to 4.
– Paths are weighted according to the discounting function $v_{sd}^{(i)} = (K - i + 1)/K$.
– Links are valued inversely proportional, i.e. $g_{sd}^{(i,l)} = btl_{sd}^{(i)}/r(l)$.
– Setting $a_{sd} = 1$ and $w = 2$, the weight of each link is then calculated by

$$w(l) = c(l) \left\{ 1 + 2 \sum_{(s,d) \in P \backslash (a,b)} \sum_{i=1}^{K} \frac{K-i+1}{K} \sum_{l \in L_{sd}^{(i)}} \frac{btl_{sd}^{(i)}}{r(l)} \right\} .$$

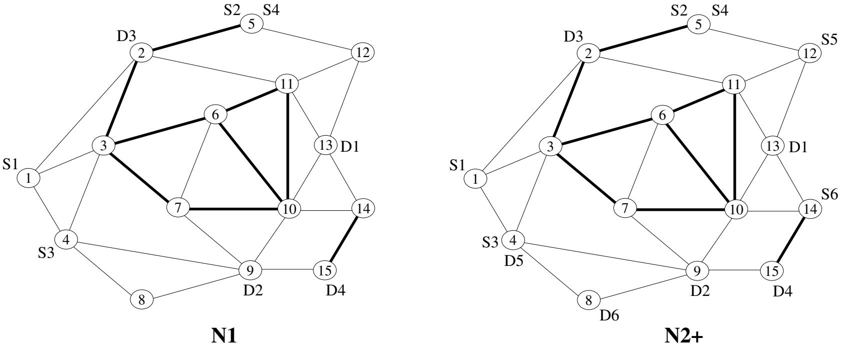


Fig. 1. Example networks N1 and N2+.

4 Numerical Results

In this section, we compare the performance of the two SMIRA-type algorithms MI-BLA and MI-PA with the shortest-path (SP), shortest-widest-path (SWP), widest-shortest-path (WSP), and, where results are available, with the S-MIRA and L-MIRA algorithms. The experiments are carried out using network topology N1 of [5], see Figure 1. Links are bi-directional with a capacity of 1200 units (thin lines) and 4800 units (thick lines).[1] Each link l is assigned a static cost $c(l)$ of one unit. The network contains the four ingress–egress pairs (S1→D1), (S2→D2), (S3→D3), and (S4→D4). Path requests are limited to those pairs only. We have chosen this network such that the performance results can be directly compared with those published in [5].

All experiments are conducted using "static" requests, i.e. the bandwidth allocated for a request is never freed again. Requests are selected randomly and are uniformly distributed among all ingress–egress pairs. In all experiments, 20 test runs were carried out, and the results shown are the mean values obtained. They have a 99% confidence interval not exceeding 1% of the mean values.

4.1 Experiment 1: Uniform Link Costs

In a first experiment, network N1 is loaded with 7000 requests. The bandwidth demand of each request is uniformly distributed in the range of 1 to 3 units (only integer values are used). Because the cost of the links in network N1 is set to 1, the SP algorithm is reduced to a minimum-hop algorithm.

The bandwidth of accepted requests of experiment 1 is shown in Figure 2. For each algorithm, the bandwidth increases with the number of requests until a saturation point is reached at which no more requests can be accommodated. The first performance measure we use is the bandwidth of successfully routed requests after the saturation point has been reached. The SP shows the weakest

[1] Owing to an error in the production of the final version of [5], links 2-5, 2-3, and 14-15 are erroneously shown as having a capacity of 1200 units. For the experiments described in [5], those links had a capacity of 4800 units.

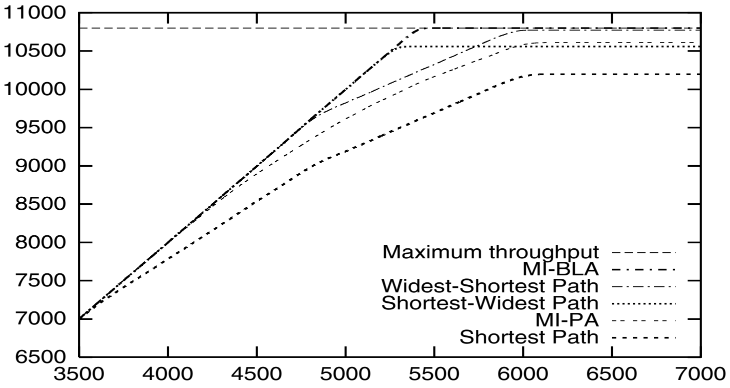


Fig. 2. Throughput of accepted requests using demands of 1 to 3 in N1.

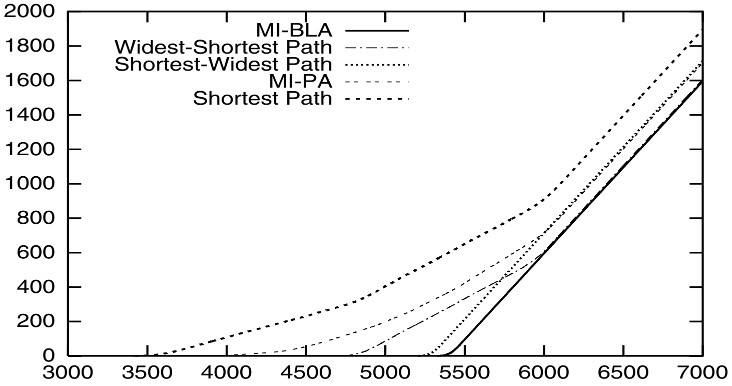


Fig. 3. Blocked requests using demands of 1 to 3 in N1.

performance, with a saturation point around 10200 bandwidth units. The best performance is shown by MI-BLA with 10800 units, followed very closely by WSP with 10770 units, and MI-PA and SWP with 10610 and 10550 units, respectively. Note that also the theoretical maximum is 10800 units. This maximum results from the solution of the multicommodity flow problem that maximizes the total flow of four commodities between the four ingress–egress pair. This is referred to as the *maximum throughput problem* [9]. Because requests are uniformly distributed among all ingress–egress pairs, we are also interested in a solution where the flow of each of the four commodities has the same value. This refers to the *maximum concurrent flow* variant of the multicommodity flow problem [9]. If the network is uniformly loaded, it is clear that a greedy routing algorithm cannot result in a flow that exceeds the maximum concurrent flow without rejecting any request. In our case, it turns out that the maximum throughput and the maximum concurrent flow have the same value of 10800. This implies that there is a solution where 2700 units can be transported between each ingress–egress pair, resulting in a total throughput of 10800 units. This maximum throughput is indicated by the "Maximum-throughput" line in Figure 2.

Table 1. Blocking rate per ingress–egress pair in N1.

Algorithm	(S1→D1)	(S2→D2)	(S3→D3)	(S4→D4)
MI-BLA	16.88%	16.81%	16.86%	16.82%
WSP	26.26%	8.42%	24.96%	8.43%
MI-PA	32.38%	8.81%	23.81%	8.79%
SWP	18.83%	18.64%	18.59%	18.59%
SP	45.25%	7.55%	25.54%	7.58%

Table 2. Number of blocked requests of total 5000 requests in N1.

Algorithm	Blocked requests			Algorithm	Blocked requests		
	Avg	Min	Max		Avg	Min	Max
Min-Hop	≈400	≈350	≈450	SP	404	353	448
WSP	≈340	≈310	≈380	WSP	86	28	151
S-MIRA	≈80	0	≈150	SWP	0	0	0
L-MIRA	0	0	0	MI-BLA	0	0	0

A second performance measure looks at the number of blocked requests. Figure 3 shows the number of blocked versus total requests. After 3450 requests, the SP algorithm starts to block some requests. MI-PA blocks after 3950 requests, WSP starts to block after 4750 requests, followed by SWP at 5230 and MI-BLA at 5350. From the above, it is clear that the blocking-free range strongly depends on the algorithm used. Note that although WSP starts to block quite early, it still achieves a throughput close to the theoretical maximum. Similarly, MI-PA has a short blocking-free range but achieves a higher total throughput than WSP does. This is due to the fact that the request-blocking rates of WSP and MI-PA differ significantly among the ingress–egress pairs. Table 1 shows the blocking rate per ingress–egress pair of various algorithms after 6500 requests have been processed, i.e. at a saturation point. MI-BLA and SWP almost achieve perfect fairness among the pairs, whereas WSP, MI-PA, and SP favor pairs 2 and 4.

Our results on the number of blocked requests are directly comparable with some of the results published in [5]. Figure 7 in [5] shows the number of blocked requests out of a total of 5000 requests for minimum-hop, WSP, S-MIRA, and L-MIRA. Table 2 compares the results presented in [5] (first four columns) with our results (columns 5 to 8). For all algorithms, the average, minimum, and maximum number of blocked requests are given.

We observe that Min-Hop closely matches our results of SP. Because uniform link costs have been used, SP actually computes minimum-hop paths. The results for WSP, however, do not match. In our experiment, WSP achieves a similar performance as S-MIRA does. Furthermore, we observe that L-MIRA achieves a perfect score with no blocked requests. In our experiment, we obtain the same result for both SWP and MI-BLA.

4.2 Experiment 2: Costs Inversely Proportional to Link Capacity

In the next experiment, we study the effect of static link costs on the performance. In network N1, all links have a cost of 1. We obtain network N2 by

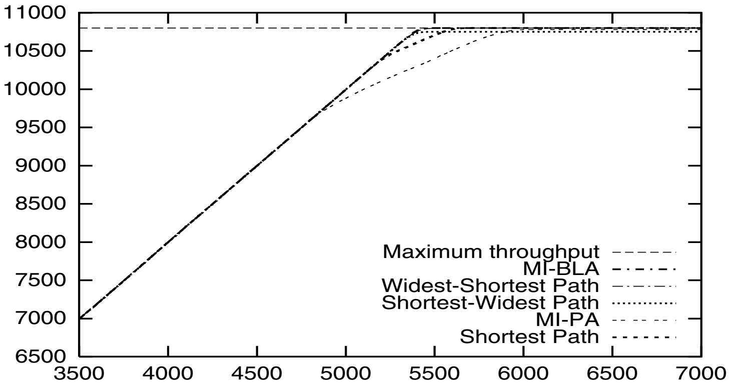


Fig. 4. Throughput of accepted requests using demands of 1 to 3 in N2.

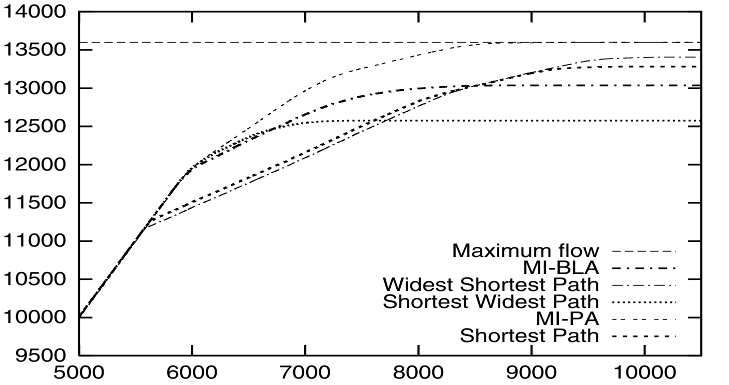


Fig. 5. Throughput of accepted requests using demands of 1 to 3 in N2+.

assigning different costs to the links. Following a common practice, we assign link costs inversely proportional to the link capacities. Links with capacity 1200 are assigned a cost of 4, and links of capacity 4800 are assigned a cost of 1. As shown in Figure 4, all algorithms perform almost equally well. The bandwidth routed by all algorithms is very close to the theoretical maximum of 10800 units. However, SP and in particular MI-PA achieve this maximum later than the other algorithms do.

4.3 Experiment 3: Additional Ingress–Egress Nodes

In a third experiment, we increase the possibility of "interference" by increasing the number of ingress–egress pairs. To obtain the example network N2+, two additional ingress–egress pairs have been added to N1, see Figure 1. A number of 11000 requests are issued, and as in the previous experiments, the requests are uniformly distributed among the six ingress–egress pairs.

Table 3. Maximum concurrent flow in N2+.

	(S1→D1)	(S2→D2)	(S3→D3)	(S4→D4)	(S5→D5)	(S6→D6)	Sum of flows
Step 1	2000	2000	2000	2000	2000	2000	12000
Step 2	2400	2000	2400	2400	2000	2000	13200
Step 3	2400	2000	2800	2400	2000	2000	13600

As shown in Figure 5, the best performance is achieved by MI-PA, reaching a total throughput of close to 13600 units. WSP and SP exhibit a very similar behavior: both start to block requests early, but are able to successfully route a total of 13400 and 13300 units, respectively. MI-BLA, on the other hand, starts to block later, but saturates earlier, at 13000 units. WSP is the least successful strategy in this environment, it reaches its saturation point already at 12600.

To compute the theoretical maximum performance of greedy algorithms, we resort to the multicommodity flow problem that maximizes the flow of six commodities corresponding to the ingress–egress pairs. In this case, it turns out that the maximum concurrent flow and the maximum throughput of the multicommodity flow problems do not coincide. In a first iteration, we find that a maximum of 2000 units of flow can be transported between each pair, resulting in a maximum concurrent flow of 12000 units. The flow can no longer be increased because some pairs are saturated. In our case it turns out that there still is residual bandwidth left between pairs (S1→D1), (S3→D3) and (S4→D4). If we compute the maximum concurrent flow in the residual network for the unsaturated pairs, we find that these pairs support another 400 units of flow. In a third step, we find that pair (S3→D3) supports another additional 400 units of flow. With this three-step approach, we obtain the maximum throughput as 13600 units. Table 3 summarizes the three maximum concurrent flow computations.

Table 3 also defines how the optimum algorithm works in the settings of experiment 3. Requests for pairs (S2→D2), (S5→D5) and (S6→D6) are blocked after 2000 units of bandwidth have been routed over those pairs. Next, request for pairs (S1→D1) and (S4→D4) are blocked after 2400 units of bandwidth. Finally, requests for pair (S3→D3) are blocked. At this point, an optimum algorithm reaches its saturation point, with 13600 units of bandwidth routed in total. For an average request size of 2, the saturation point is expected to be reached at 8400 requests.

MI-PA achieves near-optimum performance with respect to the total throughput. Figure 6 shows that also MI-PA is very close to the optimum solution with respect to the request-acceptance rate of individual pairs. MI-PA slightly over-allocates requests for pair 3 at the expense of pair 6. The request-acceptance rates of individual pairs differ significantly for WSP and SP. Compared with the optimum solution (shown on the left), both WSP and SP over-allocate requests for pairs 3 and 4, while under-allocating requests for other pairs. MI-BLA and SWP, on the other hand, show a greater fairness among the pairs.

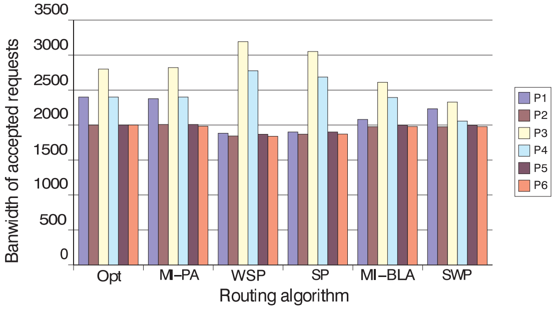


Fig. 6. Bandwidth of accepted requests per ingress–egress pair.

5 Conclusions

Here we have addressed the issue of on-line path selection for bandwidth-guaranteed requests. We have presented a new class of minimum-interference routing algorithms called "simple minimum-interference routing algorithms" (SMIRA), designed for a reduced computational complexity compared with the existing MIRA maximum-flow approach. Two typical algorithms, called MI-BLA and MI-PA, belonging to this class were introduced, and their efficiency in terms of the throughput of accepted requests and blocking-free range, as well as their fairness were assessed by means of simulation. The results obtained in the topologies considered demonstrate that these algorithms can achieve a similar optimum performance as the earlier MIRA algorithm, however, at reduced computational complexity. Comparisons with the performance of some of the established routing algorithms revealed that employment of MI-BLA and MI-PA in networks with a high degree of interference improves the performance compared with that of the shortest-path, widest-shortest-path, and shortest-widest-path algorithms. Furthermore, our algorithms exhibit a higher degree of fairness among the ingress–egress node pairs. An investigation and assessment of how the algorithms proposed perform in dynamic environments is a significant area of future work. A more systematic approach for determining the optimum algorithmic instance within the SMIRA algorithm is also a topic for further investigation.

References

1. Rosen, E., Viswanathan, A., Callon, R.: Multiprotocol Label Switching Architecture. RFC 3031 (January 2001).
2. The ATM Forum: Private Network-Network Interface Specification Version 1.0. Specification Number af-pnni-0055.000 (March 1996).
3. Ma, Q., Steenkiste, P.: On Path Selection for Traffic with Bandwidth Guarantees. In: Proc. IEEE Int'l Conf. on Network Protocols, Atlanta, GA (1997) 191-202.
4. Gawlick, R., Kalmanek, C., Ramakrishnan, K. G.: On-line Routing for Permanent Virtual Circuits. In: Proc. IEEE INFOCOM '95, Boston, MA, Vol. **1** (1998) 278-288.

5. Kar, K., Kodialam, M., Lakshman, T. V.: Minimum Interference Routing of Bandwidth Guaranteed Tunnels with MPLS Traffic Engineering Applications. IEEE J. Sel. Areas Commun. **18** (2000) 2566-2579.
6. Gibbens, R. J., Kelly, F. P., Key, P. B.: Dynamic Alternative Routing – Modelling and Behavior. In: Proc. 12th Int'l Teletraffic Congress, Turin, Italy (1988) 1019-1025.
7. Suri, S., Waldvogel, M., Warkhede, P. R.: Profile-Based Routing: A New Framework for MPLS Traffic Engineering. In: Boavida, F., Ed., Quality of Future Internet Services, LNCS **2156** (Springer, Berlin, 2001).
8. Dijkstra, E. W.: A Note on Two Problems in Connexion with Graphs. Numerische Mathematik **1** (1959) 269-271.
9. Aumann, Y., Rabani, Y.: An $O(\log k)$ Approximate Min-Cut Max-Flow Theorem and Approximation Algorithm. SIAM J. Comput. **27** (1998) 291-301.

Performance Analysis of Dynamic Lightpath Configuration for WDM Asymmetric Ring Networks

Takuji Tachibana and Shoji Kasahara

Graduate School of Information Science
Nara Institute of Science and Technology
Takayama 8916-5, Ikoma, Nara 630-0101, Japan
{takuji-t, kasahara}@is.aist-nara.ac.jp

Abstract. In this paper, we analyze the performance of a lightpath configuration method for optical add/drop multiplexer (OADM) in WDM asymmetric ring network. We consider a multiple queueing system for a node in the ring network and derive loss probability and wavelength utilization factor. Numerical examples show how arrival rate from access network and the threshold specified in the dynamic configuration method affect loss probability and wavelength utilization factor. In addition, comparing the proposed method with static configuration method, the loss probability under the proposed method can be almost the same as that under the static method where lightpaths are pre-established such that all wavelengths in the network are used efficiently.

1 Introduction

Optical add/drop multiplexer (OADM) selectively adds/drops wavelengths at any OADM to establish lightpaths in WDM network [3,4,6,7,9,11]. This provides all-optical connection between any pair of OADMs (see Fig. 1). The number of available wavelengths is 16, 32, 64, 128 and so on, and the wavelengths to be added/dropped are pre-selected in each OADM [5,8,10]. Hence significant pre-deployment network planning is required to specify what and where wavelengths are to be added/dropped. Once the network design is determined, the design will not be changed unless network operator is willing to change the network design. When the traffic pattern changes frequently, the OADM degrades the performance of network [13]. However, if wavelengths are dynamically allocated, high utilization of wavelengths and large throughput of packets are expected [2].

To realize dynamic lightpath configuration for OADM, we have proposed a dynamic lightpath configuration method [12]. With our proposed method, a lightpath is established according to the congestion state of a node and is released when there are no packets to be transmitted in a buffer for the lightpath. It is not necessary to pre-select added/dropped wavelengths.

In [12], we have considered the WDM ring network as shown in Fig. 2 where traffic is injected into each node from each access network at the same rate. Under

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 972–983, 2002.

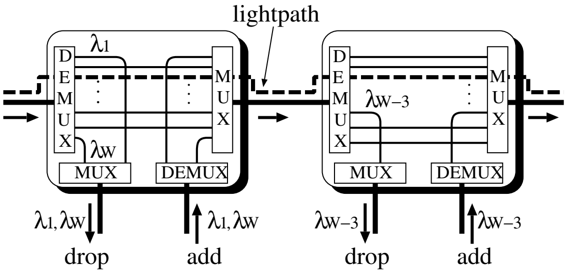


Fig. 1. Optical add/drop multiplexer.

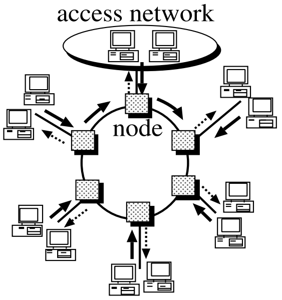


Fig. 2. Ring network model.

the real network environment, however, traffic volume injected from an access network depends on its location and services provisioned, i.e., traffic volume from each access network is different; we call such ring network an WDM asymmetric ring network.

To analyze performances of the dynamic lightpath configuration method for WDM asymmetric ring network, we further extend the symmetric ring network model in [12] to an asymmetric one. We model this system as a continuous-time Markov chain and derive the loss probability of packets coming from access network to node and wavelength utilization factor. With the analysis and simulation, we investigate how arrival rate from access network and the threshold specified in the lightpath configuration method affect the performance measures for WDM asymmetric ring network. Finally, we compare the proposed method with static configuration method and discuss the effectiveness of the proposed method.

The rest of the paper is organized as follows. Section 2 summarizes the dynamic lightpath configuration method, and in Section 3, we present the analytical model of our proposed method for WDM asymmetric ring networks. The performance analysis in the case of light traffic is presented in Section 4 and numerical examples are given in Section 5. Finally, conclusions are presented in Section 6.

2 Dynamic Lightpath Configuration Method

In this section, we summarize the dynamic lightpath configuration method proposed in [12]. Each node consists of an OADM with MPLS control plane and a label switching router (LSR) [1,2]. The procedure of lightpath configuration is as follows (see Fig. 3).

For simplicity, we consider a tandem network with three nodes, namely, nodes A, B and C. Each node is connected to its own access network through LSR. Suppose $W+1$ wavelengths are multiplexed into an optical fiber in our network. Among $W+1$ wavelengths, W wavelengths are used to transmit data traffic and one is dedicated to carry and distribute control traffic. Therefore we handle W wavelengths that consist of one default path and $W-1$ lightpaths.

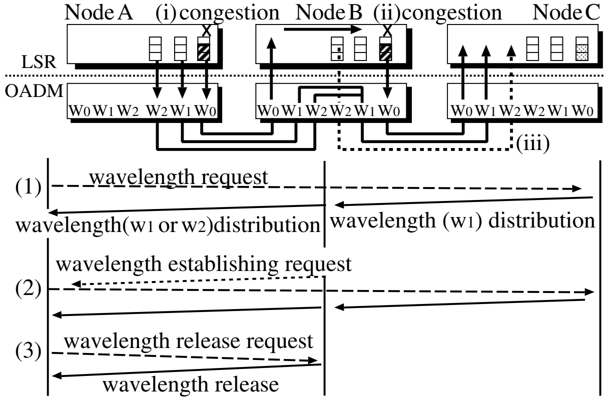


Fig. 3. Dynamic lightpath configuration.

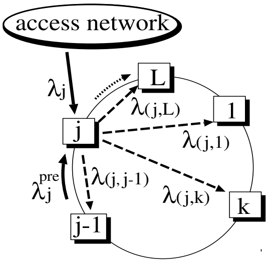


Fig. 4. Traffic from node j to other nodes.

Let w_0 denote a wavelength for default path used between adjacent nodes (A and B or B and C in Fig. 3). We define w_i $(1 \leq i \leq W - 1)$ as the wavelength which is dynamically allocated according to congestion in default path.

If an IP packet whose destination is node C arrives at node A from access network, the LSR in node A performs label switching by establishing a relation between <input port, input label> tuple and <output port, output label> tuple according to its destination node. Through MPLS control plane, OADM determines a relevant output wavelength corresponding to the output label. If the default path is not congested and a lightpath is not established between nodes A and C, the packet is transmitted to node B with wavelength w_0. When the packet arrives at node B, the LSR in node B performs label switching. Then, through MPLS control plane, the OADM in node B determines output wavelength and the packet is transmitted to node C with it.

An LSR in each node has W buffers corresponding to W wavelengths. In particular, the buffer for default path (default buffer) has pre-specified threshold. If the number of packets in default buffer becomes equal to or greater than the threshold, LSR regards the default path as being in congestion and decides to establish a new lightpath. Here the new lightpath is established between the source and destination nodes of the packet that triggers the congestion. The new lightpath is established in the following manner. Now we consider the two cases: the packet that is transmitted from nodes A to C (i) triggers congestion at node A and (ii) triggers congestion at node B.

In the case of (i), the MPLS control plane in node A requests a wavelength to the MPLS control plane in node C for the establishment of a new lightpath using control traffic (Fig. 3 (1)). Distributing network state information, MPLS control plane in each node has the latest information of lightpath configuration all the time. When the wavelength request of node A arrives at node C, MPLS control plane in node C searches an available wavelength for path BC. If wavelength

w_1 is available for path BC, node C informs node B that w_1 is available using control signal and adjusts its OADM to drop w_1.

Subsequently, the MPLS control plane in node B searches an available wavelength for path AB. If w_1 is also available for path AB, node B informs node A about it. Otherwise, node B informs node A of another wavelength, say w_2. In the latter case, w_2 is converted to w_1 at node B for the transmission from A to C. If no wavelengths are available, the new lightpath establishment fails.

Finally, node A adjusts its OADM to add w_1 or w_2. Until the lightpath establishment is completed, wavelength w_0 is still used for the packet transmission between A and C. As soon as the establishment is completed, the lightpath becomes available.

In the case of (ii) where congestion occurs at intermediate node B, the MPLS control plane in node B asks node A to request a new wavelength to node C (Fig. 3 (2)). Successive procedure is same as the case (i).

If there are no packets in the buffer after packet transmission, the timer for the holding time starts. The established lightpath is released if the holding time is over and there are no packets in the buffer (Fig. 3 (iii), (3)).

For simplicity, we assume in the paper that multiple lightpaths between any pair of nodes are not permitted.

3 System Model

We consider a WDM network where L nodes are connected in ring topology (see Fig. 2). Each node, as shown in the previous section, consists of OADM with MPLS control plane and LSR and establishes/releases lightpaths according to the dynamic lightpath configuration method. In addition, each node is connected to its own access network through LSR.

We assume that the number of wavelengths available at each node is W and all wavelengths can be converted regardless of any wavelength pairs. One of W wavelengths is for a default path and the others are for lightpaths which are dynamically established. $W-1$ wavelengths for lightpaths are numbered from 1 to $W-1$. A lightpath is established with a wavelength which has the smallest number. When there are no idle wavelength up to the $i-1$th one, a lightpath is established with the ith $(1 \leq i \leq W-1)$ wavelength.

We have two types of buffers in each node: one is for default path and the others are for lightpaths which are dynamically established/released. Let K_d denote the capacity for default buffer and K_l the capacity for each lightpath buffer. Here, the buffer capacity consists of a waiting room where packets wait for transmission and a server where a packet is in transmission. Let T_h denote the pre-specified value of threshold for default path.

For traffic condition within this WDM asymmetric ring network, we assume that packets arriving at node j $(1 \leq j \leq L)$ from access network are transmitted to destination nodes in clockwise direction. Under this assumption, we have two kinds of packet traffic that arrives at the node j: one is from the access network and the other is from the previous node $j-1$ as shown in Fig. 4.

In terms of traffic from the access network, we assume that packets arrive at node j from access network according to a Poisson process with parameter λ_j. We assume that for $1 \leq j \leq L$, λ_j is so small that packet loss hardly occurs at intermediate nodes. Moreover we assume that the destination of a packet which arrives at node j is k $(k\ /\!\!=j)$ with probability $P_k^{(j)}$ which satisfies $\sum_{\substack{k=1\\ k\ /\!\!=j}}^{L} P_k^{(j)} = 1$.

Therefore, packets sent to the destination k arrive at node j from access network according to a Poisson process with parameter $\lambda_{(j,\,k)}$ which is given by

$$\lambda_{(j,\,k)} = P_k^{(j)} \lambda_j. \tag{1}$$

Next we consider traffic which arrives at node j from node $j-1$. Since the buffers in each node are finite queues, our ring network is not an open Jackson queueing network. Due to light traffic, however, arrival packets are hardly lost and most of packets are served by default path. Therefore we can approximate the arrival process from previous node with the similar approach to the analysis of open Jackson network [14].

Let λ_j^{pre} denote the arrival rate of the packet arrival process from node $j-1$ to node j. Noting that packets are sent in clockwise direction and hardly lost due to light traffic assumption, λ_j^{pre} can be approximated with the following:

$$\lambda_j^{pre} \simeq \sum_{k=1}^{j-1} \left\{ \sum_{n=j+1}^{L} \lambda_{(k,\,n)} + \sum_{m=1}^{k-1} \lambda_{(k,\,m)} \right\} + \sum_{k=j+2}^{L} \sum_{n=j+1}^{k-1} \lambda_{(k,\,n)}, \quad 1 \leq j \leq L. \tag{2}$$

We assume that the packet arrival process at node j from the previous node $j-1$ is Poisson with rate λ_j^{pre}.

The whole packets arrive at the node j according to a Poisson process with rate $\lambda_j^{all} = \lambda_j + \lambda_j^{pre}$ which is given by

$$\lambda_j^{all} = \sum_{k=1}^{j} \left\{ \sum_{n=j+1}^{L} \lambda_{(k,\,n)} + \sum_{m=1}^{k-1} \lambda_{(k,\,m)} \right\} + \sum_{k=j+2}^{L} \sum_{n=j+1}^{k-1} \lambda_{(k,\,n)}, \quad 1 \leq j \leq L. \tag{3}$$

We define $D_l^{(j)}(t)$ as the set of destination nodes of the established lightpaths in node j at t. Then packets arrive at default path according to a Poisson process with rate $\lambda_j^{all} - \lambda_j^{light}$ where λ_j^{light} is given by

$$\lambda_j^{light} = \sum_{k \in D_l^{(j)}(t)} \lambda_{(j,\,k)}. \tag{4}$$

We also assume that for any node the transmission time of a packet, the lightpath establishment/release time and the holding time are exponentially distributed with rates μ, p and h, respectively.

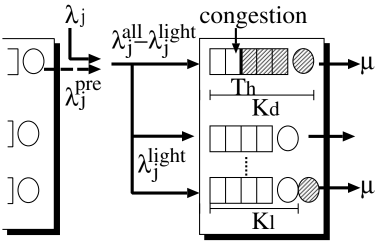


Fig. 5. Asymmetric ring node model with light traffic.

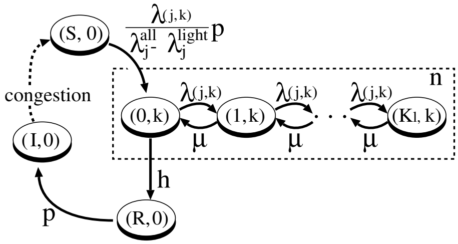


Fig. 6. State transition diagram for a lightpath $l_i^{(j)}$.

4 Performance Analysis

We consider a multiple queueing system for node j $(1 \le j \le L)$ illustrated in Fig. 5.

Let $l_i^{(j)}$ $(1 \le i \le W)$ denote the ith lightpath dynamically established/released at node j. We define the state of a lightpath $l_i^{(j)}$ $(1 \le i \le W-1)$ for node j at t as

$$J_{l_i}^{(j)}(t) = \begin{cases} n, (0 \le n \le K_l), & \text{if } l_i^{(j)} \text{ is busy,} \\ I, & \text{if } l_i^{(j)} \text{ is idle,} \\ S, & \text{if } l_i^{(j)} \text{ is being established,} \\ R, & \text{if } l_i^{(j)} \text{ is being released.} \end{cases}$$

Let $N_d^{(j)}(t)$ denote the number of packets in default path for node j at t. $d_{l_i}^{(j)}(t)$ is defined as the destination node directly connected with lightpath $l_i^{(j)}$ at t and given by

$$d_{l_i}^{(j)}(t) = \begin{cases} k, & \text{if } l_i^{(j)} \text{ is busy and connected to node } k \ (\in D_l^{(j)}(t)), \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

Finally, we define the state of the system at t as

$$(N_d^{(j)}(t),\ \boldsymbol{J}_l^{(j)}(t)), \tag{6}$$

where $\boldsymbol{J}_l^{(j)}(t)$ is given by

$$\boldsymbol{J}_l^{(j)}(t) = (\,(J_{l_1}^{(j)}(t),\ d_{l_1}^{(j)}(t)), \cdots, (J_{l_{W-1}}^{(j)}(t),\ d_{l_{W-1}}^{(j)}(t))\,). \tag{7}$$

In addition, we define $M_{l^{(j)}}^I(t)$ as the number of idle lightpaths at t, and it is expressed as

$$M_{l^{(j)}}^I(t) = \sum_{i=1}^{W-1} 1_{\{J_{l_i}^{(j)}(t)=I\}}, \tag{8}$$

where $1_{\{X\}}$ is the indicator function of event X.

Table 1. State transition rate in asymmetric ring network model.

Number of active lightpaths	Current state $(N_d,\ \boldsymbol{J}_l)$	Next state	Transition rate
$M_l^I > 0$	$N_d < T_h$	$(N_d + 1,\ \boldsymbol{J}_l)$	$\lambda_j^{all} - \lambda_j^{light}$
	$T_h \leq N_d < K_d$, $(J_{l_{i_I^{\min}}},\ d_{l_{i_I^{\min}}}) = (I,\ 0)$	$(N_d + 1,\ \boldsymbol{J}_l)$, $(J_{l_{i_I^{\min}}},\ d_{l_{i_I^{\min}}}) = (S,\ 0)$	$\lambda_j^{all} - \lambda_j^{light}$
	$N_d = K_d$, $(J_{l_{i_I^{\min}}},\ d_{l_{i_I^{\min}}}) = (I,\ 0)$	$(N_d,\ \boldsymbol{J}_l)$, $(J_{l_{i_I^{\min}}},\ d_{l_{i_I^{\min}}}) = (S,\ 0)$	$\lambda_j^{all} - \lambda_j^{light}$
	$N_d > 0$	$(N_d - 1,\ \boldsymbol{J}_l)$	μ
$M_l^I = 0$	$N_d < K_d$	$(N_d + 1,\ \boldsymbol{J}_l)$	$\lambda_j^{all} - \lambda_j^{light}$
	$N_d > 0$	$(N_d - 1,\ \boldsymbol{J}_l)$	μ

State of lightpaths	Current state $(N_d,\ \boldsymbol{J}_l)$	Next state $(N_d,\ \boldsymbol{J}_l)$	Transition rate
$(J_{l_i},\ d_{l_i}) = (S,\ 0)$	-	$(J_{l_i},\ d_{l_i}) = (0,\ k)$	$\frac{\lambda_{(j,\,k)}}{\lambda_j^{all} - \lambda_j^{light}} p$
$(J_{l_i},\ d_{l_i}) = (n,\ k)$	$n < K_l$	$(J_{l_i},\ d_{l_i}) = (n + 1,\ k)$	$\lambda_{(j,\,k)}$
	$n > 0$	$(J_{l_i},\ d_{l_i}) = (n - 1,\ k)$	μ
	$n = 0$	$(J_{l_i},\ d_{l_i}) = (R,\ 0)$	h
$(J_{l_i},\ d_{l_i}) = (R,\ 0)$	-	$(J_{l_i},\ d_{l_i}) = (I,\ 0)$	p

The state transition diagram for $l_i^{(j)}$ is illustrated in Fig. 6. Let $U^{(j)}$ denote the whole state space of $(N_d^{(j)}(t),\ \boldsymbol{J}_l^{(j)}(t))$ and $U_l^{(j)}$ the space comprised of $\boldsymbol{J}_l^{(j)}(t)$.

In the remainder of this subsection, the argument t is omitted since we consider the system in equilibrium.

The transition rate from the state $(N_d^{(j)},\ \boldsymbol{J}_l^{(j)})$ is shown in Table 1. Note that we omit the superscript (j) of any notation for the simplicity. $i_I^{\min}$ in Table 1 is defined as

$$i_I^{\min} = \min\{\, i\ ;\ J_{l_i}^{(j)} = I,\ 1 \leq i \leq W - 1\}. \tag{9}$$

For example, when current state is $(N_d^{(j)},\ \boldsymbol{J}_l^{(j)})$ where $M_{l^{(j)}}^I > 0, T_h \leq N_d^{(j)} < K_d$ and the state of $l_i^{(j)}$ is idle, a packet arrives at default path with rate $\lambda_j^{all} - \lambda_j^{light}$. Then N_d is increased by one and the lightpath establishment of $l_i^{(j)}$ starts. Similarly, when current state is $(N_d^{(j)},\ \boldsymbol{J}_l^{(j)})$ where an established lightpath $l_i^{(j)}$ has no packets in its own buffers, the holding time of $l_i^{(j)}$ is over with rate h and $l_i^{(j)}$ is released.

Let $\pi(N_d^{(j)},\ \boldsymbol{J}_l^{(j)})$ represent the steady state probability of $(N_d^{(j)},\ \boldsymbol{J}_l^{(j)})$. $\pi(N_d^{(j)},\ \boldsymbol{J}_l^{(j)})$ is uniquely determined by equilibrium state equations and following normalized condition

$$\sum_{(N_d^{(j)},\boldsymbol{J}_l^{(j)}) \in U^{(j)}} \pi(N_d^{(j)},\ \boldsymbol{J}_l^{(j)}) = 1. \tag{10}$$

Equilibrium state equations are omitted due to page limitation.

With $\pi(N_d^{(j)},\ \boldsymbol{J}_l^{(j)})$, loss probability $P_{loss}^{(j)}$ and wavelength utilization factor $P_{wave}^{(j)}$ for node j are given by

$$P_{loss}^{(j)} = \sum_{(K_d^{(j)}, \boldsymbol{J}_l^{(j)}) \in U^{(j)}} \left\{ 1 - \frac{\lambda_j^{light}}{\lambda_j^{all}} \right\} \pi(K_d^{(j)},\ \boldsymbol{J}_l^{(j)})$$
$$+ \sum_{N_d^{(j)}=0}^{K_d} \sum_{i=1}^{W-1} \sum_{d_{l_i}^{(j)} \in D_l^{(j)}} \sum_{\substack{\boldsymbol{J}_l^{(j)} \in U_l^{(j)} \\ J_{l_i}^{(j)} = K_l}} \pi(N_d^{(j)},\ \boldsymbol{J}_l^{(j)}) \frac{\lambda_{(j,\,k)}}{\lambda_j^{all}}, \tag{11}$$

$$P_{wave}^{(j)} = \sum_{(N_d^{(j)}, \boldsymbol{J}_l^{(j)}) \in U^{(j)}} \left\{ 1_{\{N_d^{(j)} > 0\}} + \sum_{i=1}^{W-1} 1_{\{(J_{l_i}^{(j)}, d_{l_i}^{(j)}) \neq (0, 0)\}} \right\} \frac{\pi(N_d^{(j)},\ \boldsymbol{J}_l^{(j)})}{W}. \tag{12}$$

5 Numerical Examples

In our numerical examples, we assume that a labeled packet size (an IP datagram + a label) is 1250 bytes within access networks and that the transmitting speed of each wavelength is 10 Gbps. Thus, the transmission speed is calculated as

$$\frac{1250 \text{ [byte]} \times 8 \text{ [bit/byte]}}{10 \text{ [Gbps]}} = 1 \ [\mu\text{s}]\,. \tag{13}$$

We set $1/\mu$ =1 [μs], where $1/\mu$ is the mean transmission time of a packet.

We set both the lightpath establishment/release time and holding time are equal to 1.0 [ms], i.e., $p = 0.001$ and $h = 0.001$. In this section, we consider WDM asymmetric ring network where there are 10 nodes.

5.1 Impact of Traffic Volume from Access Network

Figs. 7 and 8 illustrate how traffic volume from access network affects loss probability. In both figures, we set $W = 4$, $K_d = 6$, $K_l = 5$ and $T_h = 4$, and assume that the destination of each packet is equally likely, i.e., for any pair nodes j and k $(k \neq j)$,

$$P_k^{(j)} = \frac{1}{L-1} = \frac{1}{9}. \tag{14}$$

Moreover arrival rate at node 1 from access network, λ_1, is variable and other arrival rates are fixed and equal to 0.015.

Fig. 7 shows the numerical result calculated by approximation analysis and Fig. 8 represents simulation result. We observe the quantitative discrepancy between Figs. 7 and 8. This is because the loss probability in Fig. 7 is calculated under the assumption of exponential distributions of transmission time, lightpath establishment/release time and holding time while those times are set to be

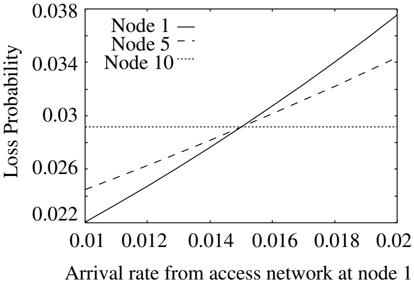


Fig. 7. Loss probability vs. arrival rate: approximation analysis.

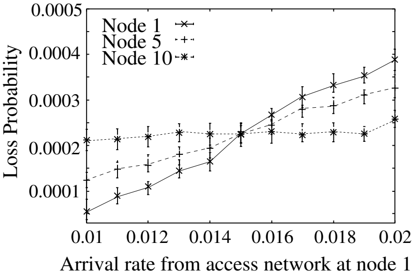


Fig. 8. Loss probability vs. arrival rate: simulation.

constant in simulation. However, both figures show the same tendency and hence our analytical model is useful for capturing the loss behavior under proposed method in a qualitative sense.

Our numerical experiments also show that our analytical model succeeds in capturing the characteristic of wavelength utilization factor, however, we omit those results due to page limitation.

In Figs. 7 and 8, we observe that loss probability for node 1 increases as arrival rate at node 1 increases while loss probability for node 10 is constant. Since destinations of packet streams originated in node 1 are equally likely, the arrival rate from previous node $j-1$, λ_j^{pre}, becomes small as the node-number j increases. This results in the small (large) loss probability if λ_1 is smaller (larger) than λ_j ($j = 2, \cdots, 10$). We further investigate this tendency in the next subsection.

5.2 Impact of Node Position

In this subsection, with our analytical result, we investigate how the loss probability and wavelength utilization factor of each node differ from those of other nodes.

Here, we set $W = 4$, $K_d = 30$, $K_l = 5$ and $T_h = 20$. The destination of each packet is equally likely, i.e., $P_k^{(j)} = 1/9$.

In terms of traffic volume from each access network, we consider the following types;

$$\text{Type A: } \lambda_i = \begin{cases} 0.18, & i = 1, 6, \\ 0.135, & \text{otherwise.} \end{cases} \qquad \text{Type B: } \lambda_i = \begin{cases} 0.09, & i = 1, 6, \\ 0.135, & \text{otherwise.} \end{cases}$$

Fig. 9 shows the loss probability against the position of node. From Fig. 9, we observe that loss probability depends on the distance from node 1 or node 6. For type A, nodes 1 and 6 have larger loss probability than others while for type B, nodes 5 and 10 have larger loss probability than others. For type A, λ_1 and λ_6 are larger than others and this makes LSRs of nodes 1 and 6 in congestion.

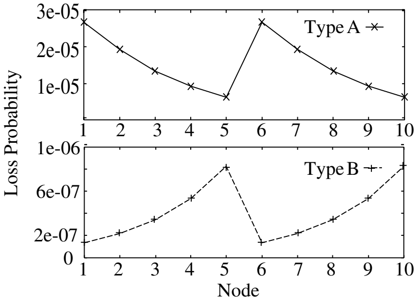


Fig. 9. Loss probability vs. node position.

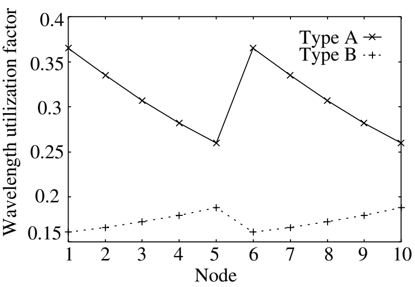


Fig. 10. Wavelength utilization factor vs. node position.

This causes the large loss probabilities of nodes 1 and 6. On the other hand, the packet streams originated in node 1 leave the ring network at nodes 2, 3, $\cdots$, and 10 in this order and hence the total arrival rate becomes small as the node-number increases. This results in the decrease of loss probabilities from nodes 2 to 5. At node 6, the same traffic volume is injected and this causes the jump of loss probability. The decrease of loss probability from nodes 6 to 10 follows from the same reason.

For type B, λ_1 and λ_6 are smaller than others and this causes the small loss probabilities at nodes 1 and 6. As the node-number increases, the traffic volume larger than λ_1 and λ_6 makes the network being congested and this results in the increase of loss probabilities at nodes from 2 (7) to 5 (10).

Fig. 10 illustrates how the proposed method establishes lightpaths in the asymmetric network. From this figure, we find that the proposed method can establish lightpaths according to traffic volume originated in each node. For type A, nodes 1 and 5 establish more lightpaths than others and for type B, nodes 6 and 10 establish more lightpaths than others. From this figure, we observe that the dynamic lightpath establishment function works well for WDM asymmetric ring network.

5.3 Impact of Threshold

In this subsection, we investigate how the threshold affects loss probability with our analytical result. We set $W = 4$, $K_d = 30$ and $K_l = 5$. As is the case with the above sections, we assume that the destination of each packet is equally likely, and consider the traffic condition for type A.

Fig. 11 shows how loss probability is affected by threshold. From Fig. 11, we observe that smaller threshold gives smaller loss probability. This is because the LSR with small threshold regards the node as being in congestion frequently and makes lightpaths busy. We also find that loss probabilities for nodes 5 and 10 do not change so much while those for nodes 1 and 6 decrease as threshold

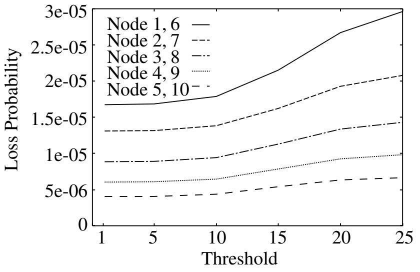


Fig. 11. Loss probability vs. threshold.

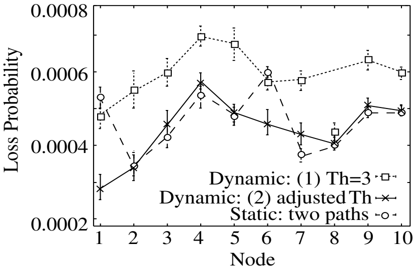


Fig. 12. Comparison of dynamic and static configurations: simulation.

becomes small. Therefore small threshold is effective to improve loss probabilities of bottleneck nodes.

5.4 Comparison of Dynamic and Static Configurations

Finally, we compare the proposed method with static configuration method where wavelengths are allocated to lightpaths statically.

Fig. 12 illustrates loss probability for each node in cases of the proposed method and static configuration method. Loss probabilities in Fig. 12 are calculated by simulation. In this figure we set $W = 4$, $K_d = 6$, $K_l = 5$ and consider the traffic condition for type A. In addition, $P_k^{(j)}$=1/9 for all j and k $(j\ /{=}k)$ except $j = 1$ and 6. For $j = 1$ and 6, we set

$$P_k^{(1)} = \begin{cases} 0.08, & k = 2, 3, \\ 0.12, & \text{otherwise.} \end{cases} \quad P_k^{(6)} = \begin{cases} 0.08, & k = 7, 8, \\ 0.12, & \text{otherwise.} \end{cases}$$

That is, more packets whose destinations are nodes 2 (7) or 3 (8) arrive at node 1 (6) than packets whose destinations are other nodes.

As for the static configuration method, we consider the case where each node statically establishes two lightpaths: one is connected to the next node and the other is connected to the next but one. Note that this is the most efficient use of wavelengths for the ring network considered here.

As for the dynamic configuration, we consider the following two cases: (1) $T_h = 3$ for all nodes and (2) T_h's are different such as

$$T_h = \begin{cases} 1, \ k = 1, 6, & 3, \ k = 3, 4, 8, 9, \\ 2, \ k = 2, 7, & 4, \ k = 5, 10. \end{cases}$$

The case of (2) is based on the results of the previous subsections.

From Fig. 12, we observe that the loss probability for dynamic configuration with $T_h = 3$ is the largest and that the loss probability for the proposed method with adjusted T_h's is almost equal to or lower than that for static configuration case. This suggests that the proposed method can establish lightpaths efficiently between pairs of nodes whose traffic volume is large.

6 Conclusion

In this paper, we have analyzed the performance of the dynamic wavelength allocation method for WDM asymmetric ring network. Numerical examples have showed that the proposed method can establish lightpaths efficiently according to traffic volume from access network even when some nodes are in congestion. In addition, the loss probability under the proposed method can be almost the same as that under the static method where lightpaths are pre-established such that all wavelengths in the network are used efficiently.

References

1. D. O. Awduche, "MPLS and Traffic Engineering in IP Networks," *IEEE Communications Magazine*, vol. 37, no. 12, pp. 42-47, Dec. 1999.
2. D. O. Awduche et al., "Multi-Protocol Lambda Switching: Combining MPLS Traffic Engineering Control With Optical Crossconnects," IETF draft-awduche-mpls-te-optical-03.txt, Apr. 2001.
3. P. Bonenfant, and A. R. Moral, "Optical Data Networking," *IEEE Communications Magazine*, vol. 38, no. 3, pp. 63-70, Mar. 2000.
4. I. Chlamtac, V. Elek, A. Fumagalli, and C. Szabó, "Scalable WDM Access Network Architecture Based on Photonic Slot Routing," *IEEE/ACM Trans. Networking*, vol. 7, no. 1, pp. 1-9, Feb. 1999.
5. O. Gerstel, R. Ramaswami, and G. H. Sasaki, "Cost-Effective Traffic Grooming in WDM Rings," *IEEE/ACM Trans. Networking*, vol. 8, no. 5, pp. 618-630, Oct. 2000.
6. N. Ghani, S. Dixit, and T. S. Wang, "On IP-over-WDM Integration," *IEEE Communications Magazine*, vol. 38, no. 3, pp. 72-84, Mar. 2000.
7. M. W. McKinnon, H. G. Perros, and G. N. Rouskas, "Performance Analysis of Boadcast WDM Networks under IP Traffic," *Performance Evaluation*, vols. 36-37, pp. 333-358, Aug. 1999.
8. Y. Miyao, "λ-Ring System: An Application in Survivable WDM Networks of Interconnected Self-Healing Ring Systems," *IEICE Trans. Commun.*, vol. E84-B, no. 6, June, 2001.
9. B. Ramamurthy, and B. Mukherjee, "Wavelength Conversion in WDM Networking," *IEEE J. Select. Areas Commun.*, vol. 16, no. 7, pp. 1061-1073, Sep. 1998.
10. R. Ramaswami, and K. N. Sivarajan, *Optical Networks: A Practical Perspective.* San Francisco: Morgan Kaufmann Publishers, 1998.
11. K. Sato, S. Okamoto, and H. Hadama, "Network Performance and Integrity Enhancement with Optical Path Layer Technologies," *IEEE J. Select. Areas Commun.*, vol. 12, no. 1, pp. 159-170, Jan. 1994.
12. T. Tachibana and S. Kasahara, "Performance Analysis of Dynamic Lightpath Configuration with GMPLS for WDM Ring Networks: The Light Traffic Case," Technical Report of IEICE (NS2001-140), pp.37-42, 2001.10.19. (in Japanese) .
13. W. Weiershausen, A. Mattheus, and F. Küppers, "Realisation of Next Generation Dynamic WDM Networks by Advanced OADM Design," *WDM and Photonic Networks*, D. W. Faulkner, and A. L. Harmer eds., IOS Press, Amsterdam, pp. 199-207, 2000.
14. R. W. Wolff, *Stochastic Modeling and the Theory of Queues.* New Jersey: Prentice Hall, 1989.

A Queueing Model for a Wireless GSM/GPRS Cell with Multiple Service Classes

D.D. Kouvatsos, K. Al-Begain, and I. Awan

Department of Computing, School of Informatics, University of Bradford
BD7 1DP, Bradford, West Yorkshire, England, UK
{d.d.kouvatsos, k.begain, i.awan}@bradford.ac.uk

Abstract. A novel analytic framework is devised for the performance modelling and evaluation of a wireless cell using Global System for Mobile telecommunication (GSM) with General Packet Radio Service (GPRS) supporting both voice and multiple class data services, respectively, under a complete partitioning scheme (CPS). In this context, a queueing model is proposed consisting of two independent queueing systems, namely an M/M/c/c loss system with Poissonian GSM traffic and a {GE/GE/1/N_1/FCFS → GE/GE/1/N_2/PS} system of access and transfer finite capacity queues in tandem having an external Compound Poisson GPRS traffic with geometrically distributed batches and generalised exponential (GE) service times under first-come-first-served (FCFS) and processor sharing (PS) scheduling rules, respectively. Although the analysis of the former loss system is straightforward, the solution of the GE-type queues in tandem is rather complex. This investigation focuses on the analysis of the tandem GE-type queueing system, which is valid for both uplink and downlink connections and provides multiple class data services with different arrival rates, interarrival-time squared co-efficient of variations (SCVs), file (burst) sizes and PS discrimination service levels. The principle of maximum entropy (ME) is used to characterise a product form approximation, subject to appropriate GE-type queueing theoretic constraints per class, and thus, implying a decomposition of the tandem system into GE/GE/1/N_1/FCFS and GE/GE/1/N_2/PS building block queues, each of which can be analysed in isolation. Subsequently, closed form expressions for state and blocking probabilities are obtained. Typical numerical examples are included to validate the ME solution against simulation and study the effect of external GPRS bursty traffic upon the performance of the cell.

Keywords: Cellular mobile system, Global System for Mobile Telecommunication (GSM), General Packet Radio Service (GPRS), wireless GSM/GPRS cell, complete partitioning scheme (CPS), performance evaluation, maximum entropy (ME) principle, generalised exponential (GE) distribution, first-come-first-served (FCFS) rule, processor sharing (PS) rule.

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 984–995, 2002.

1 Introduction

Queueing theoretic models are widely recognised as powerful and realistic tools for the performance evaluation and prediction of complex mobile systems. However, there are inherent difficulties and open issues to be resolved before a global network infrastructure for broadband mobile systems can be established. Some of these problems may be attributed to the complexity of mobile traffic characterisation and the assessment of its performance impact based on the much needed derivation of closed form metrics. Most of the published performance studies in the field are based on simulation modelling and numerical solution of Markov models covering different traffic scenarios, mostly at call level, with single or multiple service classes. Earlier proposed models are based on resource network management parameters of the Global System for Mobile Telecommunications (GSM) technology, where the capacity of radio interference in the wireless cell is divided into discrete channels and operates in circuit-switched mode (e.g., [1]. More recently, extensions of these models have been made to capture the packet-switched behaviour introduced by the General Packet Radio Service (GPRS) which has been added to GSM to allow data communication with higher bit rates than those provided by a single GSM channel (e.g., [2,3]). More recently, Foh et al [4] proposed a single server infinite capacity queue for modelling GPRS in a Markovian environment and applied matrix geometric methods for the evaluation of performance metrics.

Simulation is an efficient tool for studying detailed system behaviour but it becomes costly, particularly as the system size increases. Markov models on the other hand provide more flexibility and produce numerical results for many interesting performance measures. Nevertheless, the numerical solution of Markov models may suffer from several drawbacks, such as

- state space explosion limiting the analysis to only small mobile systems, generally consisting of one cell,
- restrictive assumptions of independent Poisson arrival processes for all types of homogeneous and uniformly distributed traffic with exponentially distributed call durations (which, if multiplexed, can be bursty and correlated).

Thus, there is still a great need to consider alternative analytic methodologies for the analysis of queueing models, based on a balanced trade-off between simplified assumptions to reduce complexity and actual real life system behaviour, leading to both credible and cost-effective approximations for the performance prediction and optimisation of mobile systems.

This investigation proposes a novel analytic framework for the performance modelling and evaluation of a wireless GSM/GPRS cell with both voice and multiple data services under a complete partitioning scheme (CPS). The work focuses on the analysis of a tandem generalised exponential (GE)-type queueing model involving a first-come-firs-served (FCFS) access queue and a discriminatory processor sharing (PS) transfer queue (air interface) with distinct multiple data service classes and external Compound Poisson GPRS (multiplexed) traffic class streams with geometrically distributed batches. The model is analysed via the principle of maximum entropy (ME) (c.f., [5,6]) which is used to characterise a product form approximation, subject to GE-type queueing theoretic constraints, and thus, allowing system decomposition and the separate analysis of each of the two GE-type queues in tandem. Subsequently, closed form expressions for state and blocking probabilities per class are obtained.

The paper is organised as follows. Section 2 describes call handling schemes for wireless GSM/GPRS cells. The GE-type tandem queueing model is discussed in Section 3 together with the characterisation of an ME product form approximation. Section 4 presents the ME analysis of the GE/GE/1/N building block queue with either FCFS or discriminatory PS scheduling rules. Numerical examples to validate the ME solution against simulation and study the effect of external GPRS bursty traffic upon the performance of the cell are included in Section 5. Concluding remarks follow in Section 6.

2 GSM/GPRS Call Handling Schemes

Resources for GPRS traffic can be reserved statically or dynamically, whereas a combination of both is possible. Different partitioning schemes can be defined where partitions are created for GSM and GPRS traffic but not for individual data services. For GPRS traffic, a complete partition is used for different data services. However, some data calls may be allocated higher priority and therefore they can be given higher share of the available bandwidth. Whenever voice and data share bandwidth, voice service is always given the highest priority. Two main partitioning schemes, namely complete partitioning and partial sharing, are described below:

- *Complete partitioning scheme (CPS)* divides the total cell capacity to serve simultaneously GSM and GPRS traffics. As a consequence, the GSM and GPRS systems can be analysed separately.
- *Partial sharing scheme (PSS)* allocates C_{data} channels for data traffic and the remaining $C_{shared} = C_{total} - C_{data}$ channels are shared by voice and data calls with preemptive priority for voice calls.

CPS has the advantage of requiring simpler management policy and implementation. Moreover, a definite capacity for GPRS under an efficient Connection Admission Control (CAC) algorithm can make feasible some QoS guarantees, although it will not clearly give the best utilisation for radio resources. Note that the CPS is the limiting case of the PSS under high loads. The GSM partition can be clearly modelled as a loss system. An admitted GSM voice call needs the assignment of a single traffic channel for its entire duration. On the absence of an available channel, a voice call is lost. Moreover, the GPRS partition can be represented by a finite capacity queueing model involving two single server queues in tandem, namely a FCFS access queue and a discriminatory PS transfer queue, where all active data connections share the total capacity of the data partition and may belong to various classes. These classes may have different characteristics such as maximum or minimum data rates, delay sensitivity, service discrimination, arrival rates, interrival-time variability and transferable file (data) length.

A transfer queue holds a finite number of data connections which are served according to a discriminatory PS rule, where the available service capacity is shared evenly amongst all data calls belonging to the same class. However, in the presence of multiple data classes with different priority levels, the service capacity is shared according to discrimination rates favouring higher priority classes. An admitted data call is initially held in a finite capacity FCFS access queue. If the access queue is full, the incoming call is lost. The access queue models the Packet Control Unit (PCU)/Sevicing GPRS Support Node (SGSN) buffers of a GPRS network with down-link traffic or the logical queue of data call request for transmission in the up-link stream. A call at the head of the access queue will be blocked if the transfer queue is full. Upon the termination of an active data call at the transfer queue, the blocked data call at the access queue is polled into the transfer queue (within a very short time required for signaling) and immediately shares in a PS fashion the available capacity.

3 The GE-Type Tandem Queueing Model

This section introduces a queueing network model for the performance analysis of a wireless GSM/GPRS cell with both voice and multiple data services under CPS. The model describes the GSM and GPRS partitions which can be studied separately (c.f., Section 2). Assuming a Poissonian arrival process, the GSM partition can be modelled by the classical Birth-and-Death M/M/c/c loss system with exponential call durations (which can be analysed via Erlang's loss formula).

The GPRS partition on the other hand can be modelled as a tandem GE-type GE/GE/1/N_1/FCFS → GE/M/1/N_2/PS finite capacity queueing system, where both external and internal traffics are approximated by GE-type interarrival-time distributions, or equivalently, Compound Poisson arrival processes, respectively, with geometrically distributed batches (c.f., Fig.1). Under PS rule, N_2 represents the maximum number of connections sharing simultaneously the available service capacity. Note that a batch arrival process is a most suitable model of bursty multiplexed connections (belonging to various classes with different minimum capacity demands) being accepted into the mobile system if there is enough service capacity at the moment of their arrival. Although the stochastic analysis of this GE-type tandem system is rather complex, the principle of maximum entropy (ME) can be used, as in earlier works [5,6], to characterise a product form approximation, subject to appropriate GE-type marginal queueing theoretic constraints.

More specifically, the form of the ME joint state probability $P(\mathbf{k}), \mathbf{k} = (\mathbf{k}_1, \mathbf{k}_2)$ of the tandem system, where $\mathbf{k}_j$ is a state vector $(k_{j1}, \ldots, k_{jR})$ and k_{ji} is the number of calls of class i in queue j for $i = 1, \ldots, R$ and $j = 1$ (access queue), 2 (transfer queue), subject to normalisation and the existence of the marginal constraints of server utilisation, mean queue length and full buffer

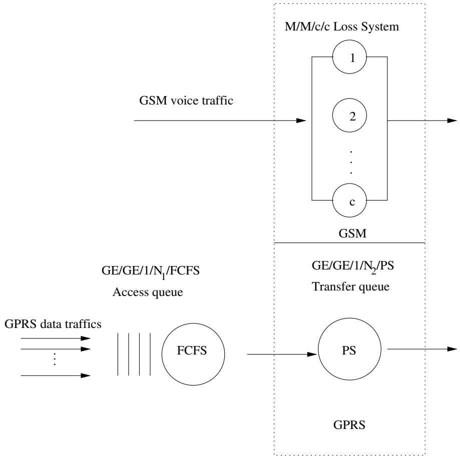


Fig. 1. The Wireless GSM/GPRS with CPS

state probability per class, can be clearly established by applying the method of Lagrange's undetermined multipliers and is given by

$$P(\mathbf{k}) = P_1(\mathbf{k}_1)P_2(\mathbf{k}_2) \tag{1}$$

where $P_1(\mathbf{k}_1)$ and $P_2(\mathbf{k}_2)$ are the marginal joint state (or, queue length) probabilities of the GE/GE/1/N$_1$/FCFS access queue and GE/GE/1/N$_2$/PS transfer queue, respectively. This product form approximation allows the decomposition of the tandem system into the two aforementioned queues, each of which can be solved in isolation by carrying out ME analysis at the queue level in conjunction with flow formulae relating, approximately, to a GE-type interdeparture-time mean and SCV (c.f., Kouvatsos et al [5]), namely

$$\lambda_{d1i} = \lambda_{1i},\, C^2_{d1i} = 2\,\langle n_{1i}\rangle\, p_{1i}(0) - C^2_{a1i}\,(1 + p_{1i}(0))\,, \tag{2}$$

where $\{\langle n_{1i}\rangle\,, i = 1, 2, \ldots, R\}$ are the marginal mean queue lengths of the access queue GE/GE/1/N$_1$/ FCFS and $\{p_{1i}(0),\, i = 1, 2, \ldots, R\}$ are the marginal probabilities that there are no data calls of class i, $i = 1, 2, \ldots, R$, in the access queue.

Note that the proposed {GE/GE/1/N$_1$/FCFS $\rightarrow$ GE/GE/1/N$_2$/PS} tandem queueing model with multiple service classes and blocking differs from and in some respect extends overall the MMPP/M/c queueing model suggested by Foh et al [4]. Although the later incorporates a PSS, Markov Modulated Poisson Process (MMPP) and multiple channels, nevertheless it is only applicable to a single service class, assumes exponential transmission times and, being an infinite capacity queueing model, does not capture the adverse effect of blocking on system performance. Moreover, the GE-type queueing models can be solved via closed form expressions as opposed to computationally expensive matrix geometric methods.

4 The ME Analysis of a GE/GE/1/N/{FCFS or PS} Queue

This section applies entropy maximisation to analyse a generic GE/GE/1/N queueing model, as a building block queue, with R (>0) classes of jobs, censored arrival processes, finite buffer capacity, N, complete buffer management scheme and either FCFS or PS service rules. Note that the GE distribution is of the form (c.f., [5,6])

$$F(t) = P(X \leq t) = 1 - \tau e^{-\tau v t}, t \geq 0 \tag{3}$$

where $\tau = 2/(C^2+1)$, X is the inter-event time random variable and $\left\{1/v, C^2\right\}$ are the mean and squared coefficient of variation (SCV) of the inter-event time distribution, respectively. Moreover, the underlying counting process of the GE distribution is a compound Poisson process with geometrically distributed batch sizes and mean batch size $1/\tau = (C^2+1)/2$.

Notation

Without the loss of generality and for the sake of simplifying the notation, the subscript j, $j = 1, 2$, referring to access and transfer queues, respectively, is dropped from the notation of this section. Let at any given time

$\mathbf{S} = (c_1, c_2, \ldots, c_n)$, $n \leq N$ be a joint system state, where c_1 is the class of the job in service and $c_\ell \in \{1, 2, \ldots, R\}$, $\ell = 2, 3, \ldots, n$ is the class of ℓ^{th} job in the queue, $\mathbf{Q}$ be the set of all feasible states of $\mathbf{S}$ and $P(\mathbf{S})$ be the stationary state probability.

For each class i, $i = 1, 2, \ldots, R$ let

λ_i be the arrival rate, μ_i be the service rate and π_i be the blocking probability that an arrival of class i finds the queue full.

For each state $\mathbf{S}$, $\mathbf{S} \in \mathbf{Q}$, and class i, $i = 1, 2, \ldots, R$, the following auxiliary functions are defined:

$$n_i(\mathbf{S}) = \text{the number of class}\, i\, \text{customers present in state}\, \mathbf{S},$$

$$s_i(\mathbf{S}) = 1, \text{if the job in service is of class}\, i \text{ or } 0, \text{ otherwise},$$

$$f_i(\mathbf{S}) = 1, \text{ if } \sum_{i=1}^{R} n_i(\mathbf{S}) = N, \text{ and}\, s_i(\mathbf{S}) = 1 \text{ or } 0, \text{ otherwise.}$$

The form of the state probability distribution, $P(\mathbf{S}), \mathbf{S} \in \mathbf{Q}$, can be characterised by maximising the entropy functional $H(\mathbf{P}) = -\sum_{\mathbf{S}} P(\mathbf{S}) \log P(\mathbf{S})$, subject to normalisation and server utilisation, mean queue length and full buffer state probability constraints per class satisfying the flow balance equations, namely

$$\lambda_i(1 - \pi_i) = \mu_i U_i, \; i = 1, \ldots, R. \tag{4}$$

By employing Lagrange's method of undetermined multipliers the following solution is obtained

$$P(\mathbf{S}) = \frac{1}{Z} \prod_{i=1}^{R} g_i^{s_i(\mathbf{S})} x_i^{n_i(\mathbf{S})} y_i^{f_i(\mathbf{S})}, \forall \mathbf{S} \in \mathbf{Q}, \tag{5}$$

where Z is the normalising constant and $\{g_i, x_i, y_i,\ i = 1, 2, \ldots, R\}$ are the Lagrangian coefficients corresponding to the server utilisation, mean queue length and full buffer state probability constraints per class, respectively. Defining the sets

$$\begin{aligned} S_0 &= \{\mathbf{S}/\mathbf{S} \in \mathbf{Q} : s_i(\mathbf{S}) = 0,\ i = 1, 2, \ldots, R\}, \\ Q_i &= \{\mathbf{S}/\mathbf{S} \in \mathbf{Q} : s_i(\mathbf{S}) = 1,\ i = 1, 2, \ldots, R\}, \\ Q_{i;\mathbf{k}} &= \{\mathbf{S} \in Q_i : n_i(\mathbf{S}) = k_i \,\&\, k_i \geq 1, i = 1, \ldots, R\}, \end{aligned}$$

and aggregating $P(\mathbf{S})$ over all feasible states $\mathbf{S} \in \mathbf{Q}$, the joint 'aggregate' state ME solution is given by

$$P(S_0) = \frac{1}{Z}, \tag{6}$$

$$\begin{aligned} P(\mathbf{k}) &= \sum_{i=1}^{R} Prob(Q_{i;\mathbf{k}}) \\ &= \frac{1}{Z} \frac{\left(\sum_{j=1}^{R} k_j - 1\right)!}{\prod_{j=1}^{R} k_j!} \left(\prod_{j=1}^{R} x_j^{k_j}\right) \left(\sum_{i=1}^{R} k_i g_i y_i^{\delta(\mathbf{k})}\right), \end{aligned} \tag{7}$$

where $\delta(\mathbf{k}) = 1$, if $\sum_i k_i = N$, or 0, otherwise, $\mathbf{k} = (k_1, k_2, \ldots, k_R)$ and k_i be the number of jobs of class i present in the queue, $i = 1, 2, \ldots, R$.

By using equations (6) and (7), closed form expressions for the aggregate state probabilities $\{P_N(n),\ n = 0, 1, \ldots, N\}$ and marginal state probabilities $\{P_i(k),\ k = 0, 1, \ldots, N_i,\ i = 1, 2, \ldots, R\}$ can be obtained (c.f., [7]). Moreover, the Lagrangian coefficients x_i and g_i can be approximated analytically by making asymptotic connections to the corresponding GE-type infinite capacity queue. Assuming x_i and g_i are invariant to the buffer capacity size N, it can be established that

$$x_i = \frac{\langle n_i \rangle - \rho_i}{\langle n \rangle},\ g_i = \frac{(1 - X)\,\rho_i}{(1 - \rho) x_i}, \tag{8}$$

where $X = \sum_{i=1}^{R} x_i$, $\langle n \rangle = \sum_{i=1}^{R} \langle n_i \rangle$ and $\langle n_i \rangle$ is the asymptotic marginal mean queue length of a multi-class GE/GE/1 queue. Note that closed form expressions for $\{\langle n_i \rangle, i = 1, 2, \ldots, R\}$ have been determined in Kouvatsos et al [5]) and are given by

$$\langle n_i \rangle = \frac{\rho_i}{2}\left(C_{ai}^2 + 1\right) + \frac{1}{2(1-\rho)} \sum_{j=1}^{R} \frac{\lambda_i}{\lambda_j} \rho_j^2 \left(C_{aj}^2 + C_{sj}^2\right),\ \{\text{for FCFS rule}\} \tag{9}$$

$$\langle n_i \rangle = \rho_i \left\{ C_{ai}^2 + \frac{1}{1-\rho} \sum_{j=1}^{R} \frac{h_j}{h_i} \rho_j C_{aj}^2 \right\},\ \{\text{for PS rule}\} \tag{10}$$

where $\rho_i = \lambda_i/\mu_i$, $\rho = \sum_{i=1}^{R} \rho_i$ and $h_i, i = 1, 2, \ldots, R$, is a set of discriminatory weights that impose service discrimination to different priority classes.

Moreover, the blocking probabilities $\{\pi_i, i = 1, 2, \ldots, R\}$ of a GE/GE/1/N queue can be approximated by focusing on a tagged data call within an arriving bulk and is determined by

$$\pi_i = \sum_{k=0}^{N} \delta_i(k)(1-\sigma_i)^{N-k} P_N(k), \tag{11}$$

where $\delta_i(k) = \frac{r_i}{r_i(1-\sigma_i)+\sigma_i}$ for $k = 0$ or 1, otherwise, $\sigma_i = 2/(1 + C^2_{a\,i})$, $r_i = 2/(1 + C^2_{s\,i})$, and $\{C^2_{a\,i}), C^2_{s\,i})\}$ are the squared coefficients of variation for the interarrival and service times per class i, respectively, $i = 1, 2, \ldots, R$.

By substituting closed form expressions for the aggregate $\{P_N(n), n = 0 \ldots N\}$ and blocking $\{\pi_i, i = 1, 2, \ldots, R\}$ probabilities into the flow balance condition (4) and after some manipulation, the following recursive relationships for the Lagrangian coefficients $\{y_i, i = 1, 2, \ldots, R\}$, can be obtained:

$$y_i^{(n)} = \left(\frac{1-\sigma_i}{X}\right) y_i^{(n-1)} - \Theta_{1i}\left(\frac{1-\sigma_i-X}{X}\right), \text{for } n \geq 2, \tag{12}$$

$$y_i^{(1)} = \Theta_{1i} - \Theta_{2i}, \tag{13}$$

where $\Theta_{1i} = \frac{1-\rho}{1-X} + \frac{\rho(1-\sigma_i)}{1-\sigma_i-X}$, and $\Theta_{2i} = (1-\sigma_i)\left(\frac{1-\rho}{1-X}\delta_i(0) + \frac{\rho}{1-\sigma_i-X}\right)$.

5 Numerical Results

This section presents some typical numerical experiments in order to illustrate the credibility of the proposed ME solution as a simple but cost-effective performance evaluation tool for assessing the effect of external GPRS traffic at the GE/GE/1/N$_1$/FCFS access queue and its propagation into the GE/GE/1/N$_2$/PS transfer queue in terms of the magnitude of the call rates and associated interarrival time squared coefficients (SCVs) of variation.

The numerical study focuses on two data service classes with different average sizes of 62.5 KBytes (class-1) and 12.5 KBytes (class-2) in conjunction with a range of corresponding SCVs, respectively. Note that these two classes may represent two typical Internet applications with different parameters, such as web browsing and email, respectively. It is assumed that the GPRS partition consists of one frequency providing total capacity of 171.2 Kbps. Among the different performance parameters that can be determined, three important ones are chosen, namely mean response time, mean queue length and blocking probability. The relative accuracy of the ME algorithm has been verified against simulation (QNAP-2 [8]) focusing on the performance measure of channel utilisation (c.f., Figs. 2-3). It can be observed that the ME results are very comparable to those obtained via simulation.

Focusing on the GE/GE/1/N/PS queue under discriminatory PS rule favouring class 1 (service discrimination weight 1:5), it can be seen that the interarrival-time SCV has an inimical effect, as expected, on the mean response time per class and the aggregate blocking probability (c.f., Figs. 4,5). Moreover, relative comparisons to assess the effects at varying degrees of interarrival time SCVs and buffer size, N, at the GE/GE/1/N/FCFS queue upon ME generated mean queue lengths are presented in Fig. 6 and 7, respectively. It can be seen that the analytically established mean queue lengths deteriorate rapidly with increasing external interarrival-time SCVs (or, equivalently, average batch sizes) beyond a specific critical value of the buffer size which corresponds to the same mean queue length for two different SCV values. It is interesting to note, however, that for smaller buffer sizes in relation to the critical buffer size and increasing mean batch sizes, the mean queue length steadily improves with increasing values of the corresponding SCVs. This ‘buffer size anomaly’ can be attributed to the fact that, for a given arrival rate, the mean batch size of arriving bulks increases whilst the interarrival time between batches increases as the interarrival time SCV increases, resulting in a greater proportion of arrivals being blocked (lost) and, thus, a lower mean effective arrival rate; this influence has much greater impact on smaller buffer sizes.

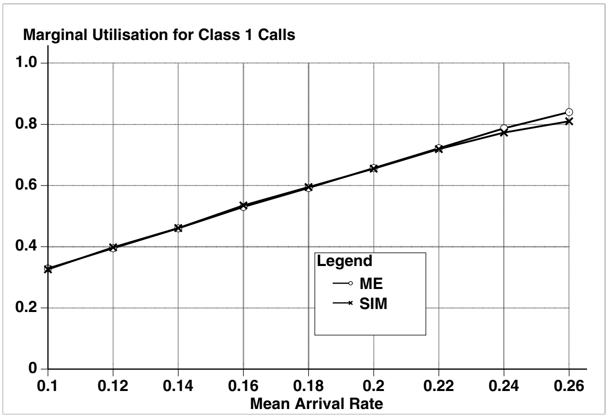


Fig. 2. Marginal Utilisations for Class 1 Calls

6 Conclusions

A novel analytic framework is presented for the performance modelling and evaluation of a wireless GSM/GPRS cell with both voice and multiple data services under a CPS, a pessimistic limiting case of the PSS. The proposed model is comprised from two independent queueing systems, namely an M/M/c/c loss system with Poissonian GSM traffic and a GE/GE/1/N_1/FCFS → GE/GE/1/N_2/PS

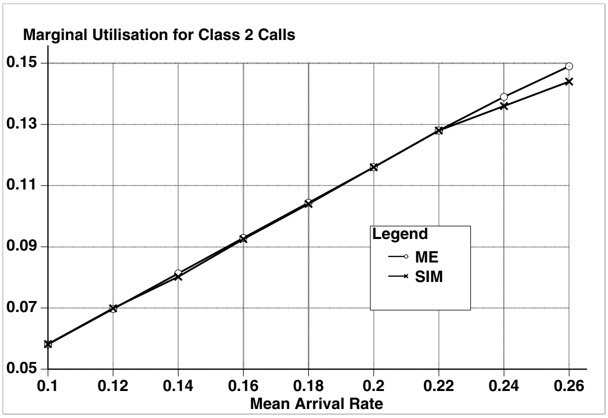


Fig. 3. Marginal Utilisations for Class 2 Calls

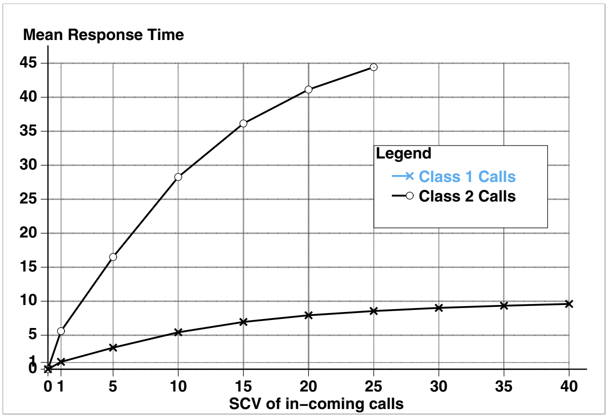


Fig. 4. Effect of varying degrees of SCV on Mean Response Time

system of access and transfer queues in tandem having a Compound Poisson external GPRS traffic with geometrically distributed batches.

The paper focuses on the analysis of the GE-type tandem system, which is valid for both uplink and downlink connections and provides voice and multiple class data services with different arrival rates and interarrival-time SCVs, file (burst) sizes and different PS discrimination service levels allowing a weighted capacity sharing. A product form approximation for the two queues in tandem is characterised, based on the principle of ME, leading into the decomposition of the system and the separate ME analysis of each building block queue under FCFS and PS rules, respectively, subject to GE-type queueing theoretic constraints per class. Subsequently, closed form expressions for state and blocking probabilities are established. Typical numerical examples are included to inves-

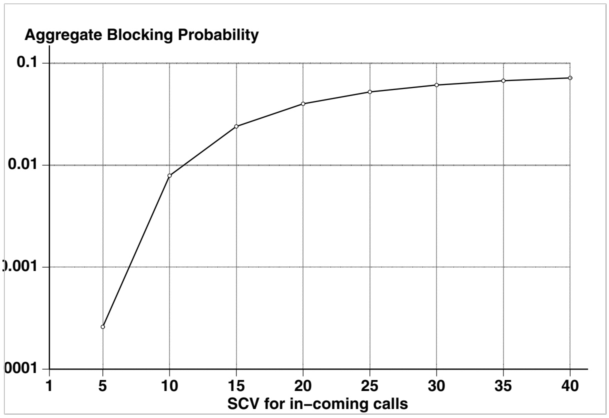


Fig. 5. Effect of varying degrees of SCV on Aggregate Blocking Probability

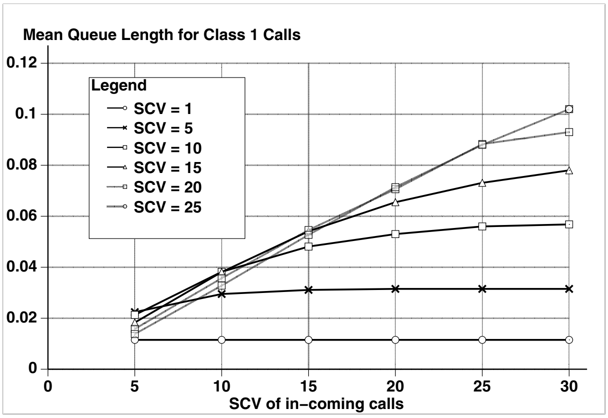


Fig. 6. Effect of varying degrees of SCV on MQLs of Class 1 at different buffer sizes

tigate the relative accuracy of the ME solution against simulation and to assess the effect of external GE-type bursty traffic upon the performance of the cell.

The paper has several extension possibilities. Firstly, the exponential assumption on the GSM call duration and interarrival time can be represented by a GE distribution resulting into a GE/GE/c/c loss system. Secondly, the model can be generalised to capture the dynamics of data partition capacity under PSS. In this case, the blocked data calls at the transfer queue will be diverted towards the loss system which will be able to accommodate R+1 classes (voice and data calls) under a preemptive resume (PR) priority rule (with voice having the highest priority). Finally, the ME methodology can be extended to model a network of multiple wireless cells using a QNM decomposition based on the principle of entropy maximisation.

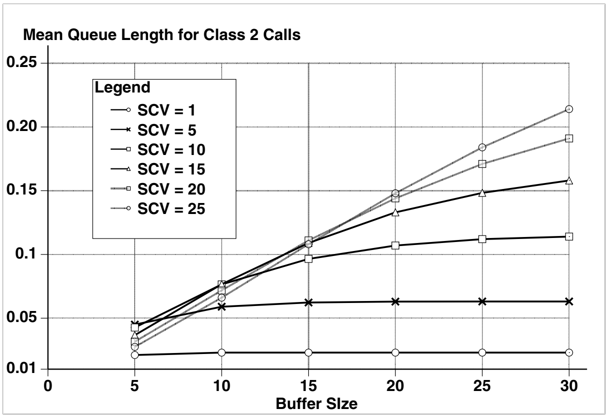


Fig. 7. Effect of varying degrees of SCV on MQLs of Class 2 at different buffer sizes

References

1. K. Begain, G. Bolch, M. Telek, Scalable Schemes for Call Admission and Handover Handling in Cellular Networks with Multiple Services. *Journal on Wireless Personal Communications*, Volume 15, No. 2, Kluwer Academic Publishers, 2000, pp. 125-144.
2. K. Begain, M.Ermel, T. Mueller, J. Schueller, M. Schweigel, Analytical Call Level Model of GSM/GPRS Network, in *SPECTS'00, SCS Symposium on Performance Evaluation of Computer and Telecommunication Systems*, Vancouver, BC, Canada, July 16-20, 2000.
3. R. Litjens, R. Boucherie, Radio Resource Sharing in GSM/GPRS Network. em ITC Specialist Seminar on Mobile Systems and Mobility, Lillehammer, Norway, March 22 - 24, 2000. pp. 261-274.
4. C.H.Foh, B.Meini, B. Wydrowski and M.Zuerman, Modeling and Performance Evaluation of GPRS, Proc. of IEEE VTC, 2001, Rhodes, Greece, pp. 2108-2112, May 2001.
5. D.D. Kouvatsos, P.H. Georgatsos and N.M. Tabet-Aouel, A Universal Maximum Entropy Algorithm for General Multiple Class Open Networks with Mixed Service Disciplines, *Modelling Techniques and Tools for Computer Performance Evaluation*, eds. R. Puigjaner and D. Potier, Plenum, pp 397-419, 1989.
6. D.D. Kouvatsos, Entropy Maximisation and Queueing Network Models, *Annals of Operation Research*, Vol. 48, pp. 63-126, 1994.
7. D.D. Kouvatsos and I.U.Awan, Open Queueing Networks with RS-Blocking and Multiple Job Classes, Research Report RR-08-01, Performance Modelling and Engineering Research Group, Department of Computing, Bradford University, August, 2001.
8. M. Veran and Potier D. QNAP-2, A Portable Environment for Queueing Network Modelling Techniques and Tools for Performance Analysis, D. Potier (ed.), North Holland, pp. 25-63, 1985.

Integrated Multi-purposed Testbed to Characterize the Performance of Internet Access over Hybrid Fiber Coaxial Access Networks

Hung Nguyen Chan, Belen Carro Martinez , Rafa Mompo Gomez, and Judith Redoli Granados

Department of Signal theory and Telematics.

Aula Cedetel - University of Valladolid.

47011 Valladolid – Spain.

hungnc@gmx.net

http://go.to/hungnc

Abstract. This paper presents an experimental testbed to study the noise effect on the performance of the transport layer over Hybrid Fiber Coaxial (HFC) networks. We have successfully designed and implemented an integrated complex testbed, which is suitable not only for laboratory environments but also reusable for real-world networks. The main purpose of the testbed is modeling the residential broadband data network using hardware simulation under several noise conditions and observing the effects on the performance of popular Internet applications as well as native TCP/UDP performances. A large number of public domain Internet measurement tools have been evaluated, from which several selective software tools have been used. In addition, new software has been developed to combine all software and hardware devices. Based on the testbed, we were able to study several issues of TCP/UDP over HFC networks by making a large number of automatic measurements and analysis. The testbed infrastructure would be very useful for cable operator and end users for monitoring and troubleshooting HFC networks, and can be effectively reused for related studies in similar environments such as wireless and DSL.

1 Motivation

Dramatic growth of the Internet has motivated the booming of broadband access technologies. Among those, Hybrid Fiber Coaxial (HFC) is one of the most popular access technologies, which provides users not only with TV programs but also high-speed Internet access and other applications. Many HFC characteristics affect the performance of Internet applications running over it, such as asymmetry, tree-and-branch topology, interferences on reverse path, etc, which requires significant considerations.

As the performance of Internet application directly reflects the cable user's satisfaction, cable operators have strong motivations to monitor not only the status of

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 996-1007, 2002.

cable modem but also the Internet application performance of users' hosts in order to effectively tackle their problems. Also, it is desirable to quickly classify and isolate the network problems. In reality, the Internet access speed of cable users depends on many factors including: RF-related factors, network congestion, QoS parameters, the load of Cable Modems Termination Systems (CMTS), etc. As a result, both users and cable operators need software tools for troubleshooting, and obtaining information about network health. This task is rather difficult without the assistance of special Internet measurement software (which most likely runs on UNIX platforms).

Regarding these issues, we have setup a multi-purposed experimental testbed to study the performance of transport layer over HFC networks. The testbed was targeted to be reused in real-world HFC networks. The additional objective of this study is to answer several questions:

[a] The possibilities of locating noise-affected area based on measuring Internet performance of hosts connected to a same CMTS (as a result, are under similar network conditions).

[b] The effect of noise on Internet applications.

[c] How to quickly distinguish between general network congestions problem and HFC network problems ?

The rest of this article is organized as follows: Section 2 provides a brief background on HFC networks. Section 3 describes the experimental testbed. In section 4, we discuss on the related studies and the contributions of this work. Several results and possible applications of the testbed are illustrated in section 5. Finally, our conclusion and future work are given in the last section.

2 Overview on HFC Networks

Figure 1 depicts a typical HFC system that provides residential broadband services. In order to bring data to cable user homes, the digital signal is converted into analog signal and mixed with CATV analog signal using frequency multiplexing. The high band 500-800 MHZ is used for downstream data and the low band from 5 to 40 MHZ is used for upstream data from cable modems.

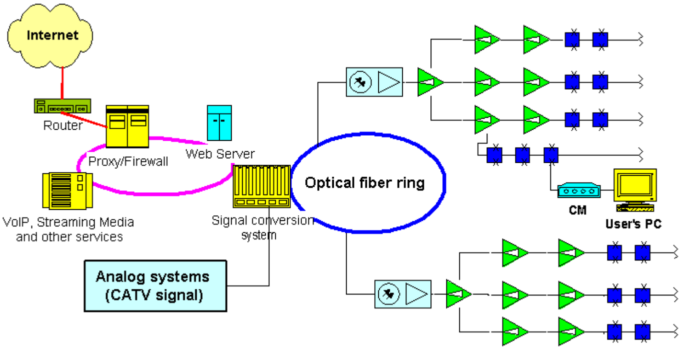


Fig. 1. A typical HFC system providing broadband data services

HFC networks are highly susceptible to noise funneling, the effect of noise entering the coaxial plant, being amplified through return path amplifiers, and aggregated from other coaxial network branches.

3 The Experimental Testbed

3.1 Hardware Configuration

Figure 2 illustrates the testbed hardware configuration. We used 6 PCs [1] running multi-operating systems including Linux RedHat 7.1, FreeBSD 4.3, Windows 2000 Server and Windows 98 SE. These OSes can be switched over from a remote PC, which controls the entire testbed. Simulated noise was generated on a PC-controlled arbitrary waveform generator HP33120 and injected into various points of the experimental network. Noise was reproduced using a noise database, taken from operating HFC networks. Another generator (HP3325B) triggered the HP33120 to control the repeated frequency of noise bursts. As a result, all noise parameters such as inter-arrival time, noise form, noise amplitude, etc, of Gaussian and impulse noise, can be fully controlled. An oscilloscope HP 54616, and a CATV analyzer HP Calan 3010R were used to monitor the signal during test. All the equipment and PCs were controlled and monitored from a remote desktop. With an additional PC-control RF switch, the connection configuration can also be changed remotely.

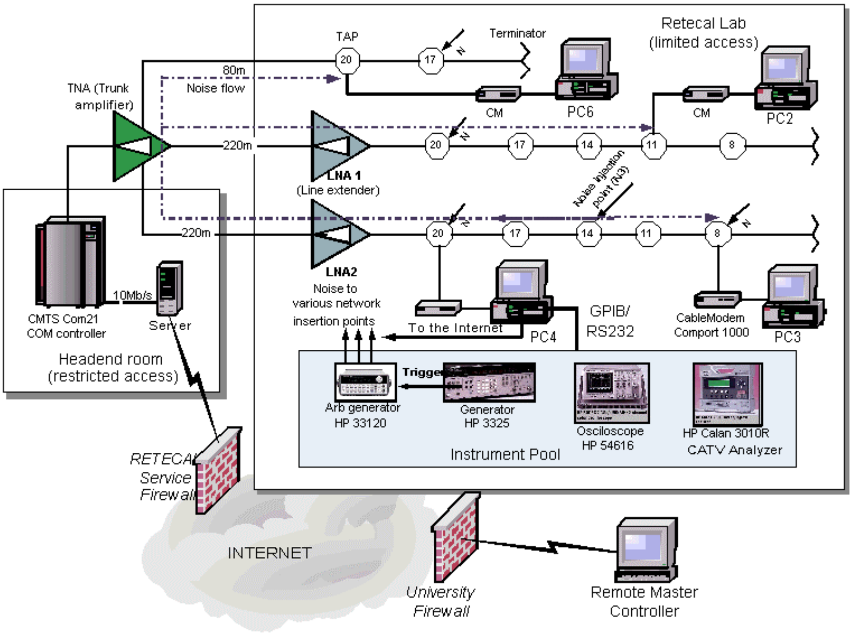


Fig. 2. The testbed hardware configuration

[1] Four PCs connected with Cable Modem have similar hardware and software configurations.

3.2 Software Configuration

The Overall Software Architecture

The testbed software structure is depicted in Figure 3. A typical client-server network providing Web and FTP service was simulated, along with a distributed measurement network. A server running Linux RedHat 7.1 provided FTP and Http services for 4 client PCs running Win98SE/LinuxRH. A remote multi-OS PC acted as master controller, controlled the whole testbed through the Internet.

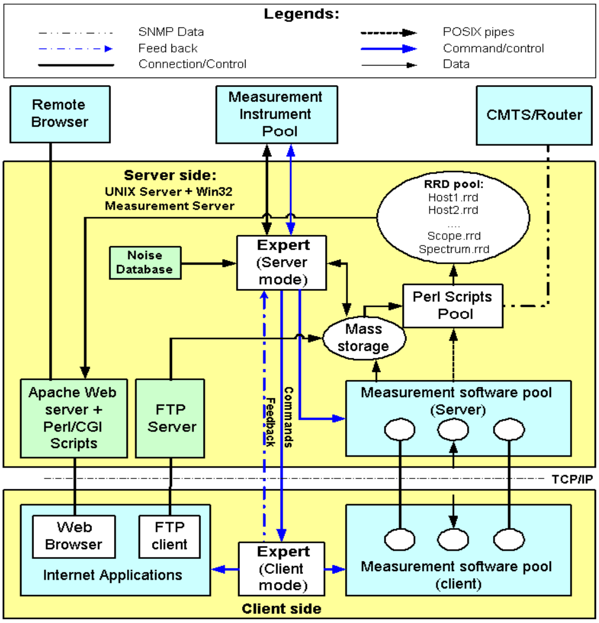


Fig. 3. Testbed software architecture

The software "Expert" plays the main role in the testbed. On the server side, a master "Expert" node can connect to a number of "Expert" slave nodes by either listening to incoming connection or actively connect to listening "Expert" slaves through direct TCP connections or SSH (Secure Socket Host) forwarding. After the communications has been successfully setup, the master node runs a master script in order to control these nodes to launch measurement programs, console commands or Internet applications (e.g. FTP, Web browser) in separated threads [2].

One "Expert" node, which acted as instrument-control server, handled a number of measurement equipments through GPIB/RS232 interfaces and saves data into a local hard disk for uploading to the Linux server. Data including log files (from servers and clients) and measurement data (from measurement instruments), is processed by various Perl scripts, and then put into a dynamically-created array of Round Robin Databases (RRD). Those scripts can also directly get run-time data from the server-

[2] Slave scripts can also run simultaneously with master script.

side measurement software pool through POSIX pipes. Another set of Perl/CGI scripts (run on top of Apache Web server) dumps selected data sources from the RRD database pool, make run-time graphs on the requests of remote Web browsers. Optionally, data from CMTS and Routers can also be acquired through SNMP interfaces and put into RRD pool.

The Main Testbed Software: "Expert"

In order to facilitate the experiment an experimental software, named "Expert", was developed. The primary objective of the software is to provide a communication links for measurement nodes and a simple graphical interface to execute/debug measurement scripts. An integrated measurement script can combine Win32 shell commands, Unix-ported commands, Windows scripting host, scripts written in scripting languages such as Perl/TCL, and a number of additional internal UNIX-style commands and communication commands. A special communication protocol was implemented in the software so that the measurement nodes can work in both client-server and peer-to-peer architecture. (More details can be found in [15], which was written by the same author.)

Moreover, the event-based feedback and the capability to control measurement instruments of the software would also be useful for the traditional HFC status monitoring on the physical layer. The design of "*Expert*" software was based on our previous experiences and codes [3], [10] and regarding related software and measurement techniques [4]. The software "Expert" was written using Visual Basic 6 on the client side and Perl 5.6 on the server side [3].

Internet Measurements

During the project, a large number of public domain tools have been evaluated on UNIX (Linux and FreeBSD) and Win32. These measurement tools are the result of many studies of the academic and OpenSource community.

Among these tools, the most useful are: 1) *Iperf* to characterize the native TCP/UDP bandwidth from cable PCs to CMTS 2) *NCS* to characterize the path from CMTSs toward the Internet backbone 3) A combination of *Tcpdump/Windump/ Tcptrace/Xplot* to analyze packet-level traces and 4) *Ntop*, a modern passive measurement software. 5) Traditional *ping*.

4 Related Studies and Contributions of This Work

At present, few studies have focused on the performance of Internet applications and protocols run over HFC since the majority of the studies ([1], [5]) focused on the physical layer in order to find solutions to detect and mitigate cable upstream interferences, which severely degrade digital services. Among these studies, S. Chaterjee [6] and P. Tzerefos [12] using OPNET, a commercial simulation software, R. Cohen [2] used NS2 simulator in order to simulate TCP-based applications over HFC network. However, these studies did not directly regard the effects of

[3] The information on the availability of the testbed software can be found on the author's home page: http://go.to/hungnc.

interferences on HFC networks. To our knowledge, one of the reasons for this is the complexity of simulating both the digital system providing Internet services and analog systems and signals only by using software simulations. Another reason is the high-cost of HFC network equipment such as CMTS, or broadband router. In addition, the HFC measurements taken on real HFC systems must be non-intrusive so as not to affect the operating network.

In the Internet measurement field, many studies have been successfully performed for decades in both traditional network environments [4], [9] as well as new environments such as wireless [13]. Numerous measurement tools have been introduced as a result of those studies.

Our previous work [3] was primarily concerned with monitoring the HFC physical layer performance by making use of a set of measurement instruments such as spectrum analyzers and oscilloscopes, which is a traditional approach. However, due to additional cost, many cable operators do not perform the preventive maintenance routines, which aim at monitoring the physical layer, even though these detailed routines have been clearly defined for years. In the next phase of our project [10], testbed software "Expert" was developed and provided automatic data collection on Win32 platform. In the current phase, server side software has been significantly improved by cross-platform Perl/CGI programs and Round Robin Database (RRD), which allow real-time display and analysis on popular Apache Web server while the client side has the new feedback capabilities.

One of our greatest challenges was the complexity of the experimental testbed. This involved a large range of issues including; controlling measurement equipment, investigating the characteristics of HFC network and Internet services from the physical layer up to the application layer, synchronizing distributed network measurement, dealing with communication with remote measurement instruments through Internet firewalls, (which eliminated the feasibility of most commercial software products such as the measurement software that implements protocols such as Agilent SICL LAN, TCP/IP VXI and MS DCOM).

In addition, most current Internet measurement tools are primarily available on UNIX platforms while the ATE (Automatic Test and measurement) software is most likely available only on Win32 platform or some proprietary platform (such as HP-UX). Our work successfully integrated Internet measurement tools with measurement equipment and software, as well as simulated typical Internet services over HFC networks. The testbed infrastructure allowed large range of studies such as the interactions between the physical layer and the link layer [11], the physical layer and the transport layer [10], and interestingly, an approach to locate noise affected area based on measuring transport layer performance, which will be very cost-effective. Several results and possible applications of the testbed will be presented in the next section.

5 Results: Possible Applications of the Testbed Infrastructure

5.1 Characterizing TCP/IP Performance over HFC under Noise Conditions

The testbed infrastructure allows studying numerous problems of HFC, such as noise funneling effects on cable user's applications, finding the vulnerable points of the networks to focus monitoring and maintenance efforts, detecting malicious-purpose noise injections, as well as the issues mentioned in section 1. Since most of the tests can be run automatically and unattended, this allows for a large number and variety of tests.

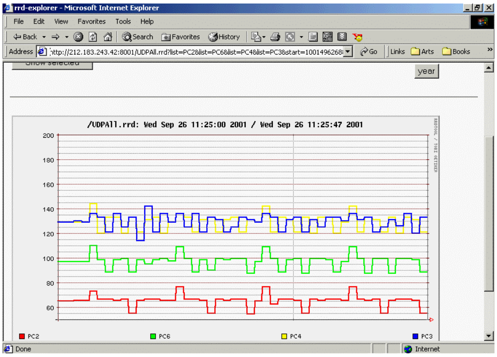

rrd-explorer - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Address :ttp://212.183.243.42:8001/UDPAll.rrd?list=PC2&list=PC6&list=PC4&list=PC3&start=1001496268
Go Links Arts Books
year
/UDPAll.rrd: Wed Sep 26 11:25:00 2001 / Wed Sep 26 11:25:47 2001
200
180
160
140
120
100
80
60
RRDTOOL / TOBI OETIKER
PC2
PC6
PC4
PC3
Done
Internet

Fig. 4. Upstream UDP throughput under noise (run-time graph)

Locating the Noise Injection Based on the Noise Isolation Factor

Based on the relative location of computers and the noise injection point shown in Figure 2, the calculation of noise isolation factor is shown in Table 1:

Table 1. Noise isolation factor

PC	Noise isolation (I) (for the connection scheme in Figure 2)
PC2	14 (tap) + cable* 3 + (- LNA2 + 220m) + 220m * 2,5db/100m (at 40MHZ) + 4 (TNA combiner loss) + LNA1 (19) + cable*4 + 11 (tap) = 54 dB
PC6	14 (tap)+ cable *3 + (- LNA2 + 220m) + 4 (TNA combiner loss) + 80m * 2,5db/100m + 20 (tap) = 40 dB
PC3	14 (tap) + cable *2 + 8 (tap) = 22 dB
PC4	14 (tap) + cable* 2 + 20 (tap) = 34 dB

While observing the table, it should be noted that the losses of short cable between taps are assumed to be zero, and (-LNA2 + 220m) = 0 dB since the testbed coaxial network was calibrated for unity gain, which means that the gain of an amplifier equals the losses that precede it. (See [7])

Figure 4 is a screenshot of a run-time graph displaying the UDP measurement results using Iperf 1.2, (with the bandwidth parameters set in accordance with the QoS parameters assigned for CMs) on 4 clients. In this figure, the noise effects on cable users can be observed. The left-most flat section corresponds with the normal operation before various impulse noise forms of increasing amplitude were injected into the network. The noise injected in point N3 (See Figure 2) causes UDP bandwidth fluctuations. It can be observed that the effects on 4 PCs are very different: PC 3 seemed not be affected until noise exceeds a threshold. The main reason is the differences of the noise isolation factors relative to the noise injection point.

We also realized that among various parameters such as round trip delay, jitter, etc, the fluctuation of UDP bandwidth is a good metric to assess the noise isolation factor, in order to predict the noise injection location.

$$\sigma = \sqrt{\frac{1}{n}\sum_{1}^{n}(x_i - \bar{x})^2} \quad \textbf{(1)}$$

$$\rho_{XY} = \frac{\mathrm{cov}(X,Y)}{\sigma_X \sigma_Y} \quad \textbf{(2)}$$

With standard deviation and correlation calculated using equation (1) and (2), respectively, the correlation between the standard deviation of UDP throughput (which represents for the UDP bandwidth fluctuation) and the isolation factor (calculated in Table 1) is -0.90338, and the correlation strength is 0.816097, which is satisfactory.

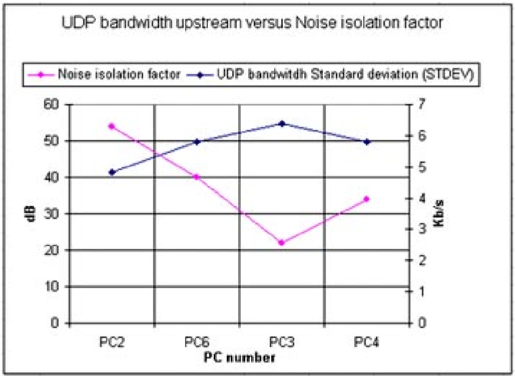


Fig. 5. Standard deviation of native UDP bandwidth versus noise isolation factor

The importance of this finding is as follows:

- Based on the estimation of noise isolation factor, together with experiences on operating networks, a vulnerable map of the network can be established (See [15]).
- If the fluctuations of UDP bandwidth are detected on a number of cable users or dedicated measurement nodes, it is most likely that noise has affected the areas that have low noise isolation relative to these measurement nodes. Typically, these areas are neighborhood taps and the taps near the bi-directional amplifiers, where upstream noise is amplified ([15]). This fact can assist in locating noise injection point, which is the solution for issue [a] mentioned in section 1.

Table 2. The correlations between noise isolation factor and various performance metrics

Performance metrics	Correlation Coefficient	Correlation Strength
Native UDP upstream	- 0,903	0,816
Native TCP upstream (measured by *Iperf*)	-0.845	0.714
FTP bandwidth upstream	-0.807	0.615
FTP bandwidth downstream	-0.796	0.633

As can be seen in table 2, the native UDP bandwidth, native TCP bandwidth, FTP bandwidth upstream and FTP bandwidth downstream are in descending order of correlation strength. This is consistent with our expectation.

Characterizing the Effects of Noise on Internet Applications

In order to investigate issue [b] in section 1, two typical Internet applications, FTP and WWW (using MSIE 5.5) were characterized in the testbed. Several results are shown on Figure 6.

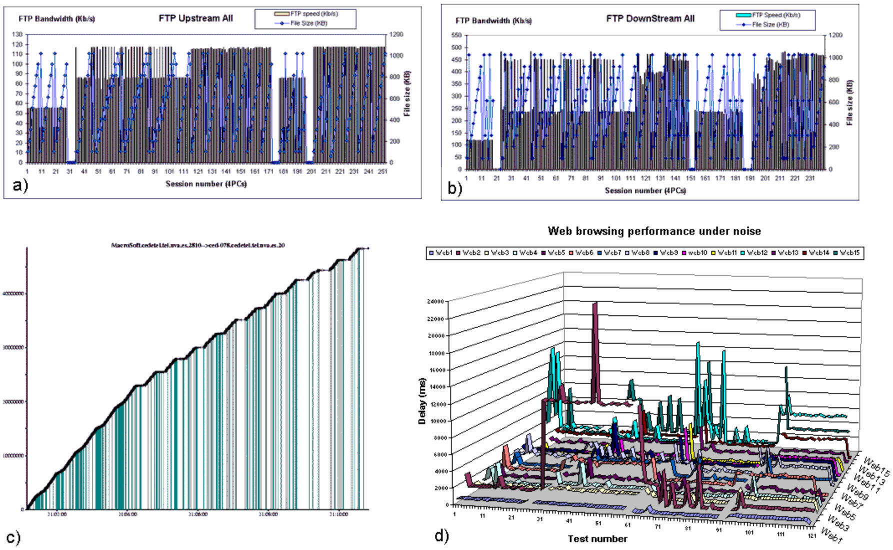


Fig. 6. Characterizing Internet applications a) FTP upstream b) FTP downstream c) A packet-level trace shows retransmissions of a FTP upstream session due to impulse noise d) Excessive Http delay and timeout due impulse noises.

As can be seen on Figure 6, under similar noise condition, the effect of ACK loss on FTP downstream throughput (Fig. 6b), can degrade throughput up to 50%, while the effect of packet loss on FTP upstream, only degrade throughput up to 25% (Fig. 6a). Web browsing is more susceptible to noise effects than FTP [4] (Figure 6d). In addition, the effects on Web browsing also heavily depend on Web page size and structure.

[4] However, in reality, users normally do not wait until a Web page is fully loaded before surfing to another page.

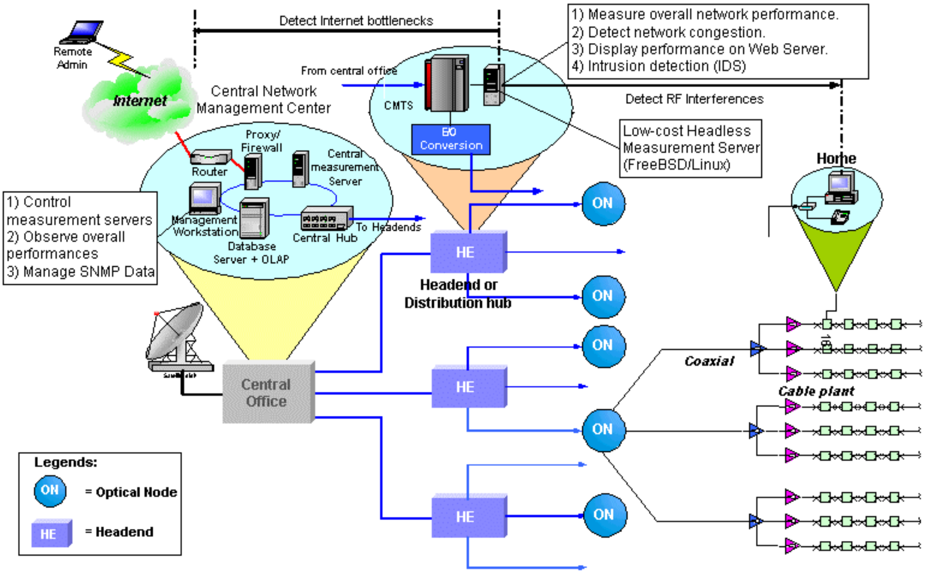

Detect Internet bottlenecks
Remote Admin
Internet
Central Network Management Center
From central office
CMTS
E/O Conversion
1) Measure overall network performance.
2) Detect network congestion.
3) Display performance on Web Server.
4) Intrusion detection (IDS)
Detect RF Interferences
Low-cost Headless Measurement Server (FreeBSD/Linux)
Home
Proxy/ Firewall
Router
Central measurement Server
Management Workstation
Database Server + OLAP
Central Hub
To Headends
1) Control measurement servers
2) Observe overall performances
3) Manage SNMP Data
HE
Headend or Distribution hub
ON
Coaxial
Cable plant
Central Office
Legends:
ON = Optical Node
HE = Headend

Fig. 7. Measurement network on HFC infrastructure

5.2 Other Applications

Monitoring HFC Network

On the next phase of the project, we are planning a measurement infrastructure based on the current testbed. Figure 6 shows the suggested measurement network, which is based on the current testbed. The measurement network will contain low-cost headless servers running FreeBSD or Linux, which are located at the network distribution centers to serve 500 to 2000 cable users. The main measurement toolset will be installed at the server while end-user software can be downloaded freely to help users monitoring and troubleshooting network health. The servers can characterize the path toward Internet backbone to detect network bottleneck as well as the path toward cable users to detect RF-related problems. Therefore, they are able to solve the issue [c] previously mentioned in section 1. In this scenario, if a software is installed on the user side, it can perform some types of measurement and diagnosis and help users to tune their network setting to obtain higher performance in HFC environments, while the measurement results can also be seen on the servers due to the client-server nature of several Internet measurement software (e.g. *Iperf*). In addition, user-side software can greatly assist in topology discovery, trouble-shooting and improve the precision of Internet measurement.

Benchmarking Cable/DSL Modems and CMTS/DSLAM

As described in [8], measurement procedures to benchmark Cable/DSL modems and/or CMTS/DSLAM can be performed by a commercial system such as Smartbit. However, in the absence of such a high-cost system, a modified testbed infrastructure can be used. Figure 7 illustrates the implementation of the testbed infrastructure as a benchmark system.

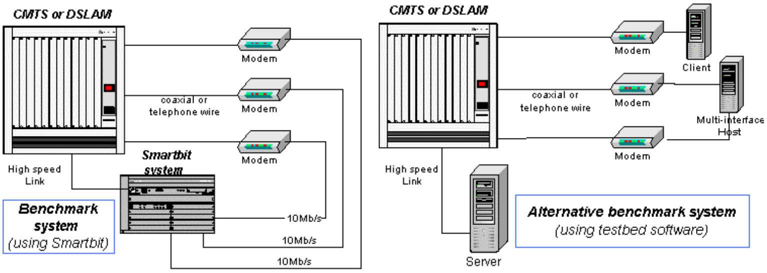


Fig. 8. Using Testbed infrastructure as a benchmark system

Other Access Network Environments (DSL, Wireless, etc)
It is widely known that DSL networks suffer from similar problems as HFC network such as noise and interferences. Moreover, since telephone wire is more susceptible to noise in comparison with coaxial, DSL performance is dramatically degraded with the presence of noise. In the very designing phase, the testbed was aimed to be transparently reused in DSL environments with very small or no modifications.

6 Conclusions and Future Work

This paper has described an experimental integrated testbed to study the overall performance of Internet access over HFC networks under simulated noise conditions and discussed on the applications of the testbed infrastructure. From the testbed experiences, a large number of measurement tools were collected and evaluated on the HFC infrastructure. The testbed infrastructure is flexible enough to be easily adapted for other purposes as well as other network environments.

One of the interesting results obtained from the experimental testbed is the novel approach of locating noise injection points based on the correlation between the noise isolation factor and the variation of UDP bandwidth. The implementation of this approach in real networks would require some network topology discovery techniques. For this purpose, there are several possibilities: 1) Obtaining ranging parameters of CMs through SNMP interfaces of CMTS. 2) Cable users can provide their CM locations themselves through client software or a database-backend Web page. 3) The network maps are already available by network designing and maintenance procedures. 4) A combinations of the above three solutions. However, the testbed still have several limitations, such as the scale of the testbed was small in comparison with real HFC networks, etc. We will address those limitations in the next phase of the project by making non intrusive-measurements on the real operating HFC networks. The server-side software will be able to assist the automatic analysis more effectively if its database capability is improved. One possibility is connecting the current RRD database pool with an OLAP engine (online analytical processing), to form multi-dimensional databases of HFC network performance.

Acknowledgements. This work is supported by *Cedetel*, Spain, and *Retecal*, a Spanish cable operator. The author also appreciates the anonymous reviewers for their comments on the previous version of this paper and the *OpenSource* community, who made many valuable software and source code available.

References

1. K.Hui Li, “Impulse noise identification for the HFC upstream Channel" IEEE transaction on broadcasting, vol. 44. No.3, pp 324-329, September 1998.
2. R. Cohen, S. Ramanathan, “TCP for high performance in hybrid fiber coaxial broadband-access networks”, IEEE/ACM Transaction on networking, pp 15-29, vol. 6. No.1, February 1998.
3. N. Chan Hung, R. Mompo, J. Redoli, B. Carro, “Flexible COM-based software solution for HFC network monitoring”, Proceeding of IFIP TC6/WG6.7, Smartnet 2000, pp 555-568, Kluwer Academic Publisher. Available: http://www.portalvn.com/hungnc/CVRevised.htm
4. CAIDA Internet measurement taxonomy. Available on http://www.caida.org
5. C. A. Eldering, N. Himayat, F. M. Gardner, “CATV Return path characterization for reliable communications”, IEEE Communication magazine, Aug 1995, pp 62-69.
6. S. Chatterjee, L. Jin , “Performance of residential broadband services over High-speed cable networks”, Proceeding of Workshop on Information Technology and Systems (WITS98), Helsinki, Finland, Dec 1998.
7. D. Raskin, D. Stoneback, “Broadband Return Systems for hybrid fiber/coax cable TV networks.”, Reading , Prentice Hall. Publishers, Inc. 1998
8. DOCSIS Acceptance Test Suit, Spirent Communication Inc., Available on Web page: http://www.spirentcom.com
9. J. Guojun , G. Yang , B. R. Crowley, D. Agarwal, “Network Characterization Service”. Available on Web page http://www-didc.lbl.gov/NCS
10. N. Chan Hung, B.Carro, R.Mompo, J.Redoli, “Monitoring the Hybrid fiber coaxial on the transport layer”. Proceeding of the European Conference on Networks and Optical communications NOC2001, pp249-256, IOS press, UK, July 2001.
11. B. Carro, N. Chan Hung, J. Redoli, R. Mompó, “Link-level effect of a noisy channel over data transmission on the return path of an HFC network”, Accepted paper for Globecom 2001, Texas USA.
12. P. Tzerefos, “On the performance and scalability of digital upstream DOCSIS 1.0 conformant CATV channels”, Ph.D. Dissertation, University of Sheffield, UK, Oct 1999.
13. R. Ludwig, A. Konrad, A. D. Joseph, “Optimizing the End-to-End performance of reliable flows over wireless links”, Proceeding of MobiCom 99.
14. N. Chan Hung, “Systematic study on Hybrid Fiber Coaxial network preventative maintenance and the performance of Internet applications over HFC networks”, Ph.D. Dissertation, University of Valladolid, Spain, Jan 2002. Available on http://www.portalvn.com/hungnc/CurrWork.htm

802.11 LANs: Saturation Throughput in the Presence of Noise[⋆]

Vladimir Vishnevsky and Andrey Lyakhov

Institute for Information Transmission Problems of RAS
B. Karetny 19, Moscow, 101447, Russia
{vishn, lyakhov}@iitp.ru
http://www.iitp.ru

Abstract. IEEE 802.11 specifies a technology for wireless local area networks (LANs) and mobile networking. In this paper, we present an analytical method of estimating the saturation throughput of 802.11 wireless LAN in the presence of noise which distorts transmitted frames. Besides the Basic Access mechanism of the 802.11 MAC protocol, we study such optional tool as the RTS/CTS method, which allows reducing the influence of collisions. In addition to the throughput, our method allows estimating a probability of a packet rejection occurring when the number of packet transmission retries attains its limit. The obtained numerical results of investigating 802.11 LANs by this method are validated by simulation and show high estimation accuracy for any values of protocol parameters and bit error rates. These results also show that the method is an effective tool for tuning the protocol parameters.

1 Introduction

In recent years, wireless data communications networks have become one of the major trends of the network industry development. Wireless LANs can be considered as an extension of the wired network with a wireless "last mile" link for connecting a large number of mobile terminals. The obvious merit of wireless LANs is the simplicity of implementation—no cables are required, its topology can be dynamically changed with connection, movement, and disconnection of mobile users without much loss of time.

The success of wireless networks depends largely on the development of networking products for multiple access to a wireless medium and of the appropriate standards. One of such standards is the IEEE 802.11 protocol [1] concerning the specifications on MAC and PHY layers for wireless networks. Leading companies (e.g., CISCO) have developed software and hardware in conformity with this standard.

The fundamental access mechanism in the IEEE 802.11 protocol is the Distributed Coordination Function (DCF), which implements the Carrier Sense

[⋆] This work was partially supported by NATO Science Programme in the Collaborative Linkage Grant PST.CLG.977405 "Wireless Access to Internet exploiting the IEEE 802.11 technology"

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 1008–1019, 2002.

Multiple Access with Collision Avoidance (CSMA/CA) method. In this method, sequential attempts to transfer by every station are separated by backoff intervals. The number of slots b in this interval is random and defined by a binary exponential backoff rule (see Section 2).

In previous works, performance of the DCF has been evaluated either by simulation (e.g., [2]) or by approximate analytical models [3,4] based on assumptions simplifying considerably the backoff rule. The DCF scheme has been studied in depth in [5]–[7], in which analytical methods have been developed for evaluating the performance of 802.11 wireless LANs in the saturation conditions when there are always queues for transmitting at every wireless LAN station. This performance index called the *saturation throughput* in [5] has been evaluated in the assumption of ideal channel conditions, i.e., in the absence of noise and hidden stations. The assumption of the absence of hidden stations is admissible as a result of the small distance between LAN stations. But if noise is neglected, the throughput may be overestimated, because electromagnetic noise in large cities is inevitable and worsens the throughput due to data distortion. In this paper we develop methods [5]–[7] to study the influence of noise on the 802.11 LAN performance.

Further in Section 2 we briefly review the DCF operation in saturation and noise. In Sections 3 and 4 we develop a new analytical method of estimating the saturation throughput and a probability of a packet rejection occurring when the number of transmission retries attains its limit. In section 5, we give some numerical research results of the saturation throughput of 802.11 LANs. These results obtained by both our analytical method and simulation allow us to validate the developed method. Finally, the obtained results are summarized in section 6.

2 DCF in Saturation

Now we briefly outline the DCF scheme, considering only the aspects that are exhibited in saturation and with absence of hidden stations. This scheme is described in detail in [1].

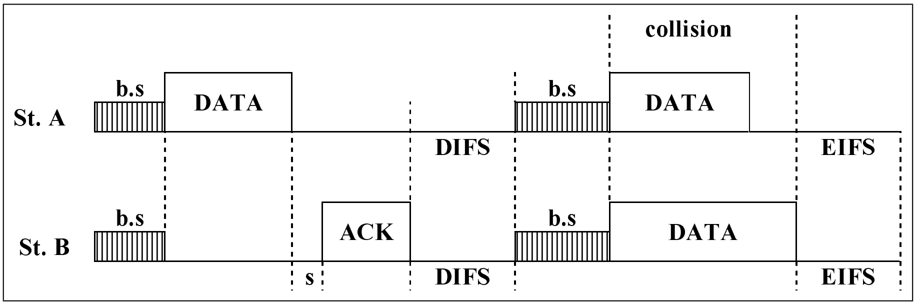


Fig. 1. Basic Access Mechanism (s - SIFS, b.s - backoff slots)

Under the DCF, data packets are transferred in general via two methods. Short packets of length not greater than $\overline{P}$ are transferred by the Basic Access mechanism. In this mechanism shown in Figure 1, a station confirms the successful reception of a DATA frame by a positive acknowledgment ACK after a short SIFS interval.

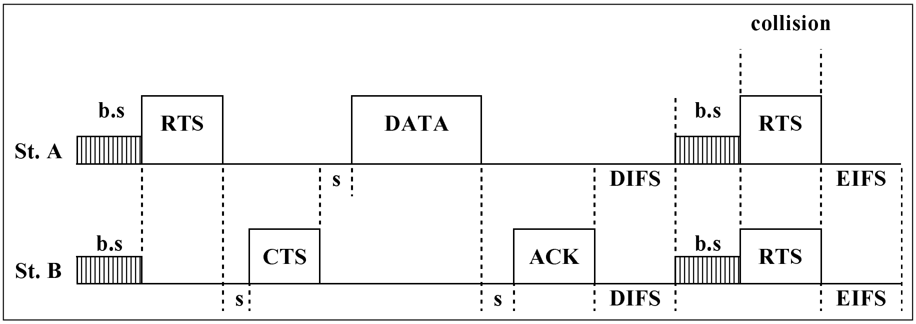


Fig. 2. RTS/CTS mechanism

Packets of length greater than the limit $\overline{P}$ called the RTS threshold in [1] are transferred via the Request-To-Send/Clear-To-Send (RTS/CTS) mechanism. In this case shown in Figure 2, first an inquiring RTS frame is sent to the receiver station, which replies by a CTS frame after a SIFS. Then only a DATA frame is transmitted and its successful reception is confirmed by an ACK frame. Since there are no hidden stations in the considered LAN, all other stations hear the RTS frame transmission and defer from their own attempts. This protects CTS, DATA and ACK frames from a collision-induced distortion. The RTS threshold $\overline{P}$ is chosen as a result of a reasonable trade-off between the RTS/CTS mechanism overhead consisting in transmitting two additional control frames (RTS and CTS) and reduction of collision duration. Figures 1 and 2 show that the collision duration is determined by the length of the longest packet involved in collision for the Basic Access mechanism, whereas in the RTS/CTS mechanism it is equal to the time of transferring a short RTS frame.

After a packet transfer attempt the station passes to the backoff state after a DIFS interval if the attempt was successful (i.e., there was no collision, all frames of a packet were transferred without noise-induced distortions) or after an EIFS interval if the attempt failed. The backoff counter is reset to the initial value b, which is called the backoff time, measured in units of backoff slots of duration σ, and chosen uniformly from a set $(0, \ldots, w-1)$. The value w, called the contention window, depends on the number n_r of attempts performed for transmitting the current packet: $w = W_{n_r}$, where

$$W_{n_r} = W_0 2^{n_r} \text{ for } n_r \leq m \text{ and } W_{n_r} = W_m \text{ for } n_r > m, \qquad (1)$$

i.e., w is equal to the minimum W_0 before the first attempt, then w is doubled after every failed attempt of the current packet transmission, reaching the maximum $W_m = W_0 2^m$. Note that every transmission attempt of a packet can include transfers of several frames (RTS, CTS, DATA, and ACK). Backoff interval is reckoned only as long as the channel is free: the backoff counter is decreased by one only if the channel was free in the whole previous slot. Counting the backoff slots stops when the channel becomes busy, and backoff time counters of all stations can decrement next time only when the channel is sensed idle for the duration of σ+DIFS or σ+EIFS if the last sensed transmission is successful or failed, respectively. When the backoff counter attains its zero value, the station starts transmission.

In the course of transmission of a packet, a source station counts the numbers of short (n_s) and long (n_ℓ) retries. Let a source station transfer a DATA frame with a packet of length equal to or less than $\overline{P}$, or an RTS frame. (Retries for these frames are called short ones in [1]). If a correct ACK or CTS frame, respectively, is received within timeout, then the n_s-counter is zeroed; otherwise n_s is advanced by one. Similarly, the n_ℓ-counter is zeroed or advanced by one in case of reception or absence of a correct ACK frame (within timeout) confirming the successful transfer of a DATA frame with a packet of length greater than $\overline{P}$ (transfer retries for that sort of DATA frames are called long retries). When any of these counters n_s and n_ℓ attains its limit N_s or N_ℓ respectively, the current packet is rejected. After the rejection or success of a packet transmission the next packet is chosen (due to saturation) with zeroing the values of n_r, n_s, and n_ℓ.

As in [5,6], to study the DCF, we adopt the following assumption: all stations change their backoff counter after a DIFS or EIFS interval closing a packet transmission attempt, i.e., the source station (or stations in case of collision), which has performed a transmission, modifies its contention window w and chooses randomly the backoff counter value from the set $(0, \ldots, w-1)$, while other stations just decrease their backoff counters by 1 (in reality [1], other stations can do it only after a backoff slot σ since the end of the DIFS or EIFS interval). Thus, at the beginning of each slot any station can start its transmission. As shown in [7], this assumption does not affect significantly the throughput estimation results with the W_0 values recommended in [1].

3 Throughput Evaluation

Let us consider a wireless LAN of N statistically homogeneous stations working in saturation. In fact, we mean by N not a number of all stations of the LAN, but a number of active stations whose queues are not empty for a quite long observation interval. By statistically homogeneity of stations, we mean that the lengths of packets chosen by every station from the queue have identical probability distribution $\{d_\ell, \quad \ell = \ell_{\min}, \ldots, \ell_{\max}\}$. Since the distance between stations is small, we assume that there are no hidden stations and noise occurs concurrently at all stations. These assumptions imply that all stations "sense" the common wireless channel identically.

As in [5], let us subdivide the time of the LAN operation into non-uniform virtual slots such that every station changes its backoff counter at the start of a virtual slot and can begin transmission if the value of the counter becomes zero.

Such a virtual slot is either (a) an "empty" slot in which no station transmits, or (b) a "successful" slot in which one and only one station transmits, or (c) a "collisional" slot in which two or more stations transmit.

As in [5,6], we assume that the probability that a station starts transmitting a packet in a given slot depends neither on the previous history, nor on the behavior of other stations, and is equal to τ, which is the same for all stations. Hence the probabilities that an arbitrarily chosen virtual slot is "empty" (p_e), "successful" (p_s), or "collisional" (p_c) are

$$p_e = (1-\tau)^N, \quad p_s = N\tau(1-\tau)^{N-1}, \quad p_c = 1 - p_e - p_s. \tag{2}$$

Thus, the throughput S is determined by the formula

$$S = \frac{p_s U}{p_e \sigma + p_s T_s + p_c T_c}, \tag{3}$$

where T_s and T_c are the mean duration of "successful" and "collisional" slots, respectively, and U is the mean number of successfully transferred data bytes in a "successful" slot.

The duration of a "collisional" slot is the sum of time of transmitting the longest frame involved in collision and an EIFS interval. Disregarding the probability of collision of three or more frames, we obtain the formula for the mean duration of a "collisional" slot

$$T_c = \sum_{\ell=\ell_{\min}}^{\overline{P}} t_d(\ell)\widehat{d_\ell}\left\{\widehat{d_\ell} + 2\left(\sum_{k=\ell_{\min}}^{\ell-1}\widehat{d_k} + \sum_{k=\overline{P}+1}^{\ell_{\max}}\widehat{d_k}\right)\right\} + t_{RTS}\left(\sum_{\ell=\overline{P}+1}^{\ell_{\max}}\widehat{d_\ell}\right)^2$$

$$+ EIFS + \delta, \tag{4}$$

where $t_d(\ell) = H + \ell/V$ is the transmission time of a DATA frame including a packet of length ℓ and a header transmitted in time H, V is the channel rate, t_{RTS} is the transfer time for an RTS frame (according to [1], $t_{RTS} < H$), and δ is the propagation delay assumed the same for all pairs of stations. Finally, $\widehat{d_\ell}$ is the probability that the performed attempt is related to a packet of length ℓ. Note that the distribution $\{\widehat{d_\ell}, \quad \ell = \ell_{\min}, \ldots, \ell_{\max}\}$ is different from the distribution $\{d_\ell, \quad \ell = \ell_{\min}, \ldots, \ell_{\max}\}$, because the longer the length of a packet, the greater the number of attempts required for transferring a packet due to the higher probability of distortion of the corresponding DATA frame by noise.

At the beginning of a "successful" slot, one and only one station initiates an attempt of transmitting a packet of length ℓ, and this transmission is successful with probability $\pi_h(\ell)$ if none of the frames exchanged between the sender and receiver in this process is distorted by noise, i.e.,

$$\pi_h(\ell) = [1 - \xi_d(\ell)](1 - \xi_a) \quad \text{for} \quad \ell \le \overline{P}$$

and

$$\pi_h(\ell) = (1 - \xi_{rc})[1 - \xi_d(\ell)](1 - \xi_a) \quad \text{for} \quad \ell > \overline{P},$$

where $\xi_{rc} = 1 - (1 - \xi_{rc})(1 - \xi_a)$ is the probability of distorting an RTS-CTS sequence by noise, while $\xi_d(\ell)$, ξ_r, and ξ_a are the probabilities of noise-induced distortion of a DATA frame including a packet of length ℓ ($\xi_d(\ell)$), and RTS (ξ_r) frame, and CTS and ACK (ξ_a) frames of identical format [1]. These distortion probabilities are defined by the Bit Error Rate (BER)—the probability of distortion of a bit, i.e., an ℓ_f-byte frame is distorted with probability $\xi_{\ell_f} = 1 - \exp\{-8\ell_f \mathrm{BER}\}$. Transfer of a packet is terminated when an exchanged frame is distorted. Thus, the mean duration of a transfer attempt in a "successful" slot depends on the length ℓ of the transferred packet and is equal to

$$t_s(\ell) = t_d(\ell) + \delta + [1 - \xi_d(\ell)](t_{ACK} + \mathrm{SIFS} + \delta)$$

$$+ \pi_h(\ell)\mathrm{DIFS} + [1 - \pi_h(\ell)]\mathrm{EIFS}$$

for $\ell \le \overline{P}$ and

$$t_s(\ell) = t_{RTS} + \delta + (1 - \xi_r)(t_{CTS} + \mathrm{SIFS} + \delta)$$

$$+ (1 - \xi_{rc})\{[1 - \xi_d(\ell)](t_{ACK} + \mathrm{SIFS} + \delta) + t_d(\ell) + \mathrm{SIFS} + \delta\}$$

$$+ \pi_h(\ell)\mathrm{DIFS} + [1 - \pi_h(\ell)]\mathrm{EIFS}$$

for $\ell > \overline{P}$, where $t_{CTS} = t_{ACK}$ is the transfer time of a CTS and an ACK frame.

Thus, the mean duration T_s of a "successful" slot and the mean number of successfully transferred bytes U in this slot are

$$T_s = \sum_{\ell=\ell_{\min}}^{\ell_{\max}} t_s(\ell)\widehat{d_\ell}, \quad U = \sum_{\ell=\ell_{\min}}^{\ell_{\max}} \ell\pi_h(\ell)\widehat{d_\ell}. \tag{5}$$

Therefore we have found all components of (3). So the throughput S can be found if the transmission commencement probability τ and the probability distribution $\{\widehat{d_\ell}\}$ are known.

4 Transmission Probability

Now we study the process of transmitting a packet of length ℓ by some station. This process starts at the instance when the packet is chosen from the queue and ends with either this packet successful transmission or its rejection. Let f_ℓ and $\overline{w}_\ell$ be the mean numbers of this packet transmission attempts and virtual slots in which the considered station defers from transmission during this process. Then

$$\tau = \sum_{\ell=\ell_{\min}}^{\ell_{\max}} d_\ell f_\ell / \sum_{\ell=\ell_{\min}}^{\ell_{\max}} d_\ell(f_\ell + \overline{w}_\ell), \tag{6}$$

$$\widehat{d_\ell} = d_\ell f_\ell / \sum_{k=\ell_{\min}}^{\ell_{\max}} d_k f_k, \quad \ell = \overline{\ell_{\min}, \ell_{\max}}. \tag{7}$$

Moreover, we will seek also the averaged probability $\overline{p}_{rej}$ of packet rejection when one of the counters n_s or n_ℓ attains its limiting value N_s or N_ℓ, respectively. This probability can be found from the following sum:

$$\overline{p}_{rej} = \sum_{\ell=\ell_{\min}}^{\ell_{\max}} d_\ell p_{rej}(\ell), \tag{8}$$

where $p_{rej}(\ell)$ is the probability of rejecting a packet of length ℓ.

In the course of transmitting a packet of length ℓ let exactly i attempts take place. Let $\psi_\ell(i)$ denote the probability of this event. Obviously,

$$\psi_\ell(i) = \psi_\ell^s(i) + \psi_\ell^r(i), \tag{9}$$

where $\psi_\ell^s(i)$ and $\psi_\ell^r(i)$ are the probabilities that this transmission process terminates at attempt i with success and rejection, respectively. In our case when exactly i attempts take place, the mean number of virtual slots in which the station defers from transmission in the course of the whole considered process is

$$\overline{W}_i = \sum_{k=0}^{i-1} \frac{W_k - 1}{2} = W_{i-1} - \frac{W_0 + i}{2}, \quad 1 \le i \le m+1,$$

$$\overline{W}_i = \sum_{k=0}^{m} \frac{W_k - 1}{2} + \frac{W_m - 1}{2}(i-1-m) = W_m \frac{i-m+1}{2} - \frac{W_0 + i}{2}, \quad i > m+1.$$

Then we have

$$f_\ell = \sum_{i=1}^{i_m(\ell)} i\psi_\ell(i), \quad \overline{w}_\ell = \sum_{i=1}^{i_m(\ell)} \overline{W}_i \psi_\ell(i), \tag{10}$$

where $i_m(\ell)$ is the maximal number of attempts for such a packet, i.e., $i_m(\ell) = N_s$ for $\ell \le \overline{P}$ and $i_m(\ell) = i_m^1 = (N_s - 1)N_\ell + 1$ for $\ell > \overline{P}$.

Now we look for probabilities $\psi_\ell(i)$. First we consider a simple case $\ell \le \overline{P}$, in which the number i of attempts is bounded by N_s. The probability of unsuccessful attempt is

$$\pi_{cd}(\ell) = 1 - (1 - \pi_c)(1 - \xi(\ell))$$

where

$$\pi_c = 1 - (1-\tau)^{N-1} \quad \text{and} \quad \xi(\ell) = 1 - (1 - \xi_d(\ell))(1 - \xi_a)ø$$

are the probabilities of the current attempt collision and distorting DATA or ACK frames, respectively. Then the process is completed successfully at the ith attempt with probability

$$\psi_\ell^s(i) = [1 - \pi_{cd}(\ell)][\pi_{cd}(\ell)]^{i-1}, \quad i = 1, \ldots, N_s, \tag{11}$$

or ends in rejection with probability

$$p_{rej}(\ell) = [\pi_{cd}(\ell)]^{N_s}, \tag{12}$$

i.e.,

$$\psi_\ell^r(i) = 0 \quad \text{for} \quad i < N_s \quad \text{and} \quad \psi_\ell^r(N_s) = [\pi_{cd}(\ell)]^{N_s}. \tag{13}$$

Consequently by (9),

$$\psi_\ell(i) = [1-\pi_{cd}(\ell)][\pi_{cd}(\ell)]^{i-1}, \ \ i = 1, \ldots, N_s - 1, \ \ \psi_\ell(N_s) = [\pi_{cd}(\ell)]^{N_s - 1}. \tag{14}$$

Now let $\ell > \overline{P}$. In this case the number i_d of DATA frame transfer attempts is bounded by N_ℓ and each of these attempts may be preceded by $0, \ldots, N_s - 1$ unsuccessful attempts of transferring an RTS frame. Moreover, in the case of a packet rejection due to attaining the limit N_s, the packet transmission process completes with N_s failed RTS transfer attempts.

Let us express the probability $\psi_\ell^r(i)$ as the sum

$$\psi_\ell^r(i) = p_{rej}^d(\ell, i) + p_{rej}^r(\ell, i), \tag{15}$$

where $p_{rej}^d(\ell, i)$ and $p_{rej}^r(\ell, i)$ are the probabilities of rejection after i packet transmission attempts due to the attainment of limiting values of the n_ℓ- and n_s-counters, respectively. Note that

$$p_{rej}(\ell) = \sum_{i=1}^{i_m^1} [p_{rej}^d(\ell, i) + p_{rej}^r(\ell, i)]. \tag{16}$$

The probabilities of unsuccessful transfer of DATA and RTS frames are $\xi(\ell)$ and $\pi_{cr} = 1 - (1 - \pi_c)(1 - \xi_{rc})$, respectively. Therefore after simple algebraic operations we obtain

$$\psi_\ell^s(i) = (1-\pi_{cr})[1-\xi(\ell)]\pi_{cr}^{i-1} \sum_{h=0}^{\min(i, N_\ell)-1} \left(\frac{\rho_\ell}{\pi_{cr}}\right)^h g(i-1-h, h+1), \quad i = 1, \ldots, i_m^1, \tag{17}$$

$$\begin{aligned} p_{rej}^d(\ell, i) &= 0, \quad i = 1, \ldots, N_\ell - 1, \\ p_{rej}^d(\ell, i) &= \pi_{cr}^{i-N_\ell} \rho_\ell^{N_\ell} g(i - N_\ell, N_\ell), \quad i = N_\ell, \ldots, i_m^1, \\ p_{rej}^r(\ell, i) &= 0, \quad i = 1, \ldots, N_s - 1, \quad p_{rej}^r(\ell, N_s) = \pi_{cr}^{N_s}, \end{aligned} \tag{18}$$

$$p_{rej}^r(\ell, i) = \pi_{cr}^i \sum_{h=1}^{\min(i-N_s, N_\ell - 1)} \left(\frac{\rho_\ell}{\pi_{cr}}\right)^h g(i - N_s - h, h), \quad i = N_s + 1, \ldots, i_m^1, \tag{19}$$

where $\rho_\ell = (1 - \pi_{cr})\xi(\ell)$ is the probability that an attempt of transmitting a packet of length ℓ fails just due to noise-induced distortion of DATA or ACK frames, while $g(u, v)$ is the number of ways in which u indistinguishable balls (failed RTS transfer attempts) can be placed in v urns (gaps preceding each of DATA transfers) so that every urn contains not more than $N_s - 1$ balls. The function $g(u, v)$ is computed recursively:

$$g(0, v) = 1 \quad \forall v > 0, \quad g(u, 1) = 1 \text{ for } u < N_s \text{ and } 0 \text{ for } u \geq N_s,$$

$$g(u,v) = \sum_{k=0}^{\min(u,N_s-1)} g(u-k,v-1) \quad \text{for} \quad v \geq 2, \quad u > 0.$$

Therefore the transmission probability τ can be estimated by the following iterative procedure.

Step 0. Define an initial value for τ.

Step 1. For all possible packet lengths ℓ and number of attempts i, compute the rejection probabilities $\psi_\ell(i)$ by (14) if $\ell \leq \overline{P}$ or by (9), (15), and (17)–(19) if $\ell > \overline{P}$.

Step 2. For all possible packet lengths ℓ, using (10), compute the mean numbers of attempts f_ℓ and virtual slots $\overline{w}_\ell$ in which transmission is postponed.

Step 3. Using (6), find the modified value of τ and compare it with the initial value. If the difference of these values is greater than a predefined limit, return to Step 1 using a new initial value for τ—the half-sum of its old initial value and the modified value.

After this iterative procedure, we obtain the averaged rejection probability $\overline{p}_{rej}$ by (8), (12), (16), (18), and (19). Finally, we find the distribution $\{\widehat{d_\ell}\}$ by (7) and throughput by the formulas of the previous section.

We don't prove exactly the convergence of this iterative technique due to its complexity and lack of space. It is clear intuitively that the equation (6) has a unique solution because a growth of transmission probability τ leads to increasing the collision probability and, hence, to increasing the average number $\overline{w}_\ell/f_\ell$ of slots anticipating an attempt for all ℓ. In practice, numerous examples of adopting the suggested technique with various values of wireless LAN parameters have shown that this technique provides very fast convergence to the solution and high speed of calculating the values of estimated performance indices. It takes less than a second to calculate S and $\overline{p}_{rej}$ with running this technique program implementation at Intel Celeron 400 MHz.

5 Numerical Results

To validate our model, we have compared its results with that obtained by GPSS (General Purpose Simulation System) simulation [8]. The object of our numerical investigations was a LAN which consisted of N statistically homogeneous stations working in saturation and was controlled by the DCF scheme of the IEEE 802.11 protocol with the higher-speed physical layer extension (802.11b) [9]. The values of protocol parameters used to obtain numerical results for the analytical model and simulation were the default values [9] for the Short Preamble mode and summarized in Table 1. Moreover, the information packet size ℓ (in bytes) is sampled uniformly from the set $\{1, \ldots, 1999\}$.

In our simulation model, we have tried to take into account of all real features of the 802.11 MAC protocol and, of course, not adopted the assumptions used with analytical modeling and described at the end of Section 2 and in Section 3. In the course of each run (it took about 2 hours, in average) of the simulation model, we watched the measured performance index value and stopped the simulation when this value fluctuations became quite small (within 0.5%).

Table 1. Values of protocol parameters

Slot time, σ	20 μs	Propagation time, δ	1 μs
MAC+PHY Header	49 bytes	Length of ACK and CTS	29 bytes
Header transfer time, H	121 μs	ACK transfer time, t_{ACK}	106 μs
RTS length	35 bytes	RTS transfer time, t_{RTS}	111 μs
SIFS	10 μs	DIFS	50 μs
EIFS	212 μs	V	11 Mbps
Short retry limit, N_s	7	Long retry limit, N_ℓ	4
Minimal contention window, W_0	32	Maximal contention window, W_m	1024

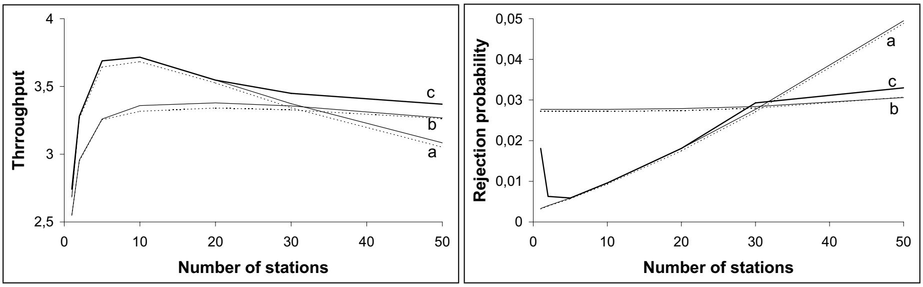


Fig. 3. Throughput (Mbps) and rejection probability versus number of station with BER= $5 \cdot 10^{-5}$ for (a) the Basic Access mechanism, (b) the RTS/CTS mechanism, and (c) the optimal hybrid mechanism

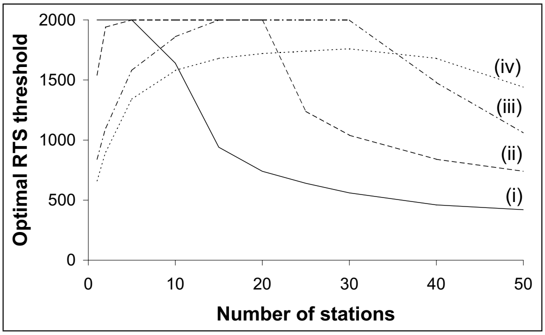


Fig. 4. Optimal RTS threshold (bytes) versus number of station with (i) BER= $1 \cdot 10^{-5}$, (ii) BER= $5 \cdot 10^{-5}$, (iii) BER= $1 \cdot 10^{-4}$, and (iv) BER= $1.4 \cdot 10^{-4}$

In Figure 3, we present some results of studying the throughput and the averaged rejection probability for the Basic Access and RTS/CTS mechanisms (where $\overline{P} > l_{max}$ and $\overline{P} = 0$, respectively) with varying the number N of stations. Here dotted curves have been obtained by simulation, while our method has been adopted to obtain other curves. First of all, let us note a high accuracy of the analytical model: the errors never exceed 2% with throughput estimation and 5% with rejection probability estimation.

Further, as we could expect, the Basic Access mechanism provides the highest throughput when a number N of stations is small ($N < 30$ in Figure 3), while the RTS/CTS mechanism is better when N is large and provides nearly the same throughput with increasing the number of stations.

The bold curves in Figure 3 have been obtained for the hybrid mechanism with the optimal RTS threshold $\overline{P}_{opt}$ providing the maximal throughput and depending on N. The optimizing curves are shown in Figure 4 for various values of BER and have been determined with our analytical method. (A high calculation speed of our method has allowed us to use the exhaustive search of the optimal threshold.) With a low BER (curve (i)), $\overline{P}_{opt}$ is quite small for large N, increases monotonically with decrease of N until some threshold N_b (where $\overline{P}_{opt}$ becomes equal to $l_{max} + 1 = 2000$ bytes), and remains constant with $N \leq N_b$, that is, the Basic Access mechanism is the best for small N. With a high BER (curves (ii)–(iv)), a curve $\overline{P}_{opt}(N)$ is not monotonic, that is, an additional threshold N_b^0 appears somewhere below N_b and $\overline{P}_{opt}$ decreases with decrease of N from N_b^0 to 1. For example, $N_b = 30$ and $N_b^0 = 15$ with BER$= 1 \cdot 10^{-4}$. Both thresholds increase with BER growth, but N_b^0 increases faster so that the interval, where the Basic Access mechanism is the best, disappears and these thresholds unite into one with a very high BER (see curve (iv)).

Thus, we have obtained the following surprising fact: when stations are few and a BER is high, the best mechanism is not the Basic Access one, but some hybrid mechanism, and the throughput improvement achieved by this optimization is significant. For example, with $N = 2$ and BER$= 1 \cdot 10^{-4}$, $S = 1.44$ Mbps for the Basic Access mechanism and $S = 1.62$ Mbps for the optimal hybrid mechanism with $\overline{P} = \overline{P}_{opt} = 1100$ bytes.

This case of few stations in a LAN can seem “exotic” and negligible, but keeping in mind that we considered only active stations, it corresponds to a real-life situation of low traffic. As Figure 3 shows, the throughput improvement in the considered case is achieved at the expense of worsening a rejection probability: in the above example, $\overline{p}_{rej} = 0.057$ for the Basic Access mechanism and $\overline{p}_{rej} = 0.131$ for the optimal hybrid mechanism. It can be explained in the following way. When stations are few and a BER is high, a collision probability is small and a failure probability is equal approximately to a noise-induced distortion probability for a DATA frame. So we can assume that a maximal number of attempts of transmitting a packet is equal to $N_d = 4$ if the packet is transmitted by the RTS/CTS mechanism and to $N_s = 7$ with the Basic Access mechanism. For a given packet and BER, the less maximal number of attempts, the larger the rejection probability, the less the mean value of backoff intervals anticipating transmission attempts, and hence the larger the throughput.

6 Conclusions

In this paper, a continuation of [5]–[7], a simple analytical method is developed for estimating the throughput of a wireless LAN controlled by the DCF scheme of IEEE 802.11 protocol and operating under saturation and in noise. Besides the throughput, the probability of a packet transfer rejection due to the attainment of the limiting values specified by the Standard [1] for the number of attempts for transferring long and short frames is evaluated. According to numerical results, our method is quite exact and can be considered as an effective tool for both investigating the influence of bit error rate on the wireless LAN performance indices and tuning optimally the protocol parameters.

Extensions of the developed method to take into account of a possible presence of hidden stations as well as to consider the real-life situations when traffic generated by wireless LAN stations is non-uniform and non-saturating seem possible and are proposed as a future research activity. In order to tackle new research issues generated by the use of wireless LANs as Internet access networks, we plan also to apply the results of studying the 802.11 MAC layer for investigating the interaction between this protocol and the TCP/IP protocol stack (i.e., the protocols of Internet).

References

1. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. ANSI/IEEE Std 802.11, 1999 Edition.
2. Weinmiller, J., Schlager, M., Festag, A., et al.: Performance Study of Access Control in Wireless LANs – IEEE 802.11 DFWMAC and ETSI RES 10 HYPERLAN. Mobile Networks and Applications **2** (1997) 55–76
3. Chhaya, H.S. and Gupta, S.: Performance Modeling of Asynchronous Data Transfer Methods of IEEE 802.11 MAC Protocol. Wireless Networks **3** (1997) 217–234
4. Ho, T.S. and Chen, K.C.: Performance Analysis of IEEE 802.11 CSMA/CA Medium Access Control Protocol. Proc. 7th IEEE Int. Symp. on Personal, Indoor and Mobile Radio Communications (PIMRC'96), Taipei, Taiwan (1996) 407–411
5. Bianchi, G.: Performance Analysis of the IEEE 802.11 Distributed Coordination Function. IEEE Journal on Selected Areas in Communications **18** (2000) 535–548
6. Calí, F., Conti, M., and Gregory, E.: Dynamic Tuning of the IEEE 802.11 Protocol to Achieve a Theoretical Throughput Limit. IEEE/ACM Transactions on Networking **8** (2000) 785–799
7. Vishnevsky, V.M. and Lyakhov, A.I.: IEEE 802.11 Wireless LAN: Saturation Throughput Analysis with Seizing Effect Consideration. Cluster Computing **5** (2002) 133–144
8. T.J. Schriber: Simulation using GPSS. John Wiley & Sons (1974)
9. Higher-Speed Physical Layer Extension in the 2.4 GHz Band. Supplement to [1]

Efficient Simulation of Blocking Probabilities for Multi-layer Multicast Streams

Jouni Karvo

Networking Laboratory, Helsinki University of Technology,
P.O.Box 3000, FIN-02015 HUT, Finland.
Jouni.Karvo@hut.fi

Abstract. This paper presents an efficient algorithm for Monte-Carlo simulation of time blocking probabilities for multi-layer multicast streams with the assumption that blocked calls are lost. Users may join and leave the multicast connections freely, thus creating dynamic multicast trees. The earlier published algorithms are applicable to small networks or networks with few users. The present simulation algorithm is based on the inverse convolution method, and is the only effective way to handle large systems, known to the author.

1 Introduction

This paper presents an efficient algorithm for Monte-Carlo simulation of time blocking probabilities for multi-layer multicast streams with the assumption that blocked calls are lost. Consider a network with circuit switched traffic, or packet switching with strict quality guarantees, such as the IntServ architecture in the Internet. Decisions on whether to allow a new connection in the network are made according to availability of resources.

In general, traffic is a mixture of point-to-point (unicast) and point-to-multipoint (or multicast) traffic. There are well known algorithms for calculating blocking probabilities for unicast traffic in absence of multicast traffic, see e.g. [1, 2]. Multicast, however, gives rise to a multitude of new problems, (see e.g. [3]), one of which is blocking probability calculation. A model called “multicast loss system” has been developed for calculating blocking probabilities in recent years. This system comprises a tree-structured multicast network with dynamic membership. In this network, users at the leaf nodes can join or leave any of the several multicast channels offered by one source, the root of the tree. The users joining the channels form dynamic multicast connections that share the network resources. Blocking occurs when there are not enough resources available in the network to satisfy the resource requirements of a request. Blocked calls are lost. The multicast loss system may be seen as a virtual network over the real one, carrying the multicast traffic of the real network.

The time blocking probability is the probability that the system is in a state where a call cannot be established due to unavailable resources, while the call blocking probability is the probability that a user’s attempt to establish a call

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 1020–1031, 2002.

fails due to unavailable resources. These probabilities are intimately related, and it is possible to calculate the call blocking probability in a multicast loss system knowing the time blocking probability.

Audio and video streams can be coded hierarchically [4]. In hierarchical, or layered, coding, information is separated according to its importance, and then coded and transmitted in separate streams. In the present setting a user may, depending on her needs and abilities, subscribe to the most important sub-stream only, in which case she is said to be on layer 1, or subscribe to any number r of the most important sub-streams, in which case she is on layer r. This paper studies the effective simulation of blocking probabilities for multicasted layered streams. The assumption that blocked calls are lost implies that if a user does not get the desired layer (or number of sub-streams) due to blocking, she will not get any layer. That is, there will be no re-negotiation of lower quality transmission.

Chan and Geraniotis [5] studied the system of layered video multicasting. They gave the definition of the state space, but resorted to approximations for the actual calculations. After their work, research has concentrated on non-layered multicast streams, see e.g. [6,7]. An efficient Monte-Carlo simulation method for dynamic multicast networks with single layer multicast streams has been developed by Lassila et al. [8]. This method was based on the inverse convolution method Lassila and Virtamo published in [9]. Recently, there has been progress in the case where the multicast streams are layered. Karvo et al. [10] developed an algorithm for calculating blocking probabilities of two-layer streams with Poisson arrivals and exponential holding times. They extended their study in [11] to an arbitrary number of layers, and studied the validity of the insensitivity property for different user models. The present paper provides an efficient simulation algorithm extending the inverse convolution approach of Lassila et al. [8] to this multi-layer case.

This paper is organized as follows. Section 2 presents the basic system model, and the time blocking probability calculation with exponential computational complexity. The problem of estimating time blocking probabilities is divided into smaller sub-problems in section 3. Section 4 contains the main contribution of this paper, showing how the inverse convolution method is applied to the layered multicast case. A numerical example is given in section 5, and the results are summarised in section 6.

2 Multicast Loss System

This section presents the system model and the notation for the multicast loss system. This model is the same as in [11]. Consider a network consisting of J links, indexed with $j \in \mathcal{J} = \{1, \dots, J\}$, link j having a capacity of C_j resource units. The network is organized as a tree. The set $\mathcal{U}$ denotes the set of user populations, located at the leaves of the tree. The leaf links and the user populations connected to them are indexed with the same index $u \in \mathcal{U} = \{1, \dots, U\}$. The set of links on the route from user population u to the root node is denoted by $\mathcal{R}_u$. The user populations downstream link j, i.e. for which link $j \in \mathcal{R}_u$, are denoted by $\mathcal{U}_j$. The size of the set $\mathcal{U}_j$ is denoted by U_j. Let $\mathcal{M}_j$ denote the set of all links downstream link j (including link j), and $\mathcal{N}_j$ the set of neighbouring links downstream link j (excluding link j). The links of the tree are indexed so that for all $j' \in \mathcal{N}_j$, $j' < j$. Thus, the root link is denoted by J. The multicast network supports I channels, indexed with $i \in \mathcal{I} = \{1, \ldots, I\}$. The channels originating from the root node represent different multicast transmissions, from which the users may choose. There are L layers. Each layer $l \in \mathcal{L} = \{1, \ldots, L\}$ has a capacity requirement of $d(l)$ capacity units. The capacity requirements are unique and $d(l) < d(l')$ for all $l < l'$, i.e. layer L contains all hierarchically coded sub-streams, layer 2 the two most important ones, and layer 1 only contains the most important sub-stream.

2.1 State Space

The states of the channels in a link define the state of that link. Each channel is in one of the states $\{0, 1, \ldots, L\}$, depending on whether the channel is *off*, or on layer $1, \ldots, L$. That is, the state of channel i on link j is $Y_{j,i} \in \{0, \ldots, L\}$. The vector $\mathbf{Y}_j = (Y_{j,i}; i \in \mathcal{I}) \in \{0, \ldots, L\}^I$ denotes the state of link j. The tuple (u, i, l) of the user population u (leaf link), channel i and layer l defines a multicast connection. The states $\mathbf{Y}_u$ of all the leaf links define the network state $\mathbf{X}$,

$$\mathbf{X} = (\mathbf{Y}_u; u \in \mathcal{U}) = (Y_{u,i}; u \in \mathcal{U}, i \in \mathcal{I}) \in \Omega\,, \tag{1}$$

where $\Omega = \{0, \ldots, L\}^{U \times I}$ denotes the network state space. The network state determines the state of any link j as follows:

$$\mathbf{Y}_j = \begin{cases} \mathbf{Y}_u, & \text{if } j = u \in \mathcal{U}\,, \\ \max\limits_{u' \in \mathcal{U}_j} (\mathbf{Y}_{u'}), & \text{otherwise}\,, \end{cases} \tag{2}$$

where $\max(\cdot)$ denotes the component-wise max-operation. The occupancy of any link j is determined by the link state as

$$S_j = D(\mathbf{Y}_j) = \sum_{i=1}^{I} d(Y_{j,i})\,, \tag{3}$$

where $d(0) = 0$, i.e. when channel is *off*, it does not need any link capacity. The occupancy generated by all other channels but I is denoted by $S'_j = D'(\mathbf{Y}_j) = \sum_{i=1}^{I-1} d(Y_{j,i})$.

Finally, in a finite capacity network, the capacity constraints of the links truncate the state space,

$$\tilde{\Omega} = \left\{ \mathbf{x} \in \Omega \,\middle|\, D(\mathbf{y}_j) \leq C_j, \forall j \in \mathcal{J} \right\}. \tag{4}$$

2.2 Probability Distributions

Let us assume that the user populations of the leaf links are independent, and that the leaf link distributions $\pi_u(\mathbf{y}) = \mathrm{P}\{\mathbf{Y}_u = \mathbf{y}\}$, $u \in \mathcal{U}$, are known, and represent stationary distributions of reversible Markov processes satisfying the detailed balance equations. Several types of user population models of this kind have been discussed in [7], and in [11].

The steady state probabilities $\pi(\mathbf{x})$ of the network states in a system with infinite link capacities can be calculated from

$$\pi(\mathbf{x}) = \mathrm{P}\{\mathbf{X} = \mathbf{x}\} = \prod_{u \in \mathcal{U}} \pi_u(\mathbf{y}_u)\,, \tag{5}$$

since the user populations are independent. The inverse convolution approach also dictates that all channels shall be independent. Thus,

$$\pi_u(\mathbf{y}_u) = \prod_{i \in \mathcal{I}} p_{u,i}(y_{u,i})\,. \tag{6}$$

As already noted in [11], probabilities $\tilde{\pi}(\mathbf{x})$, $\mathbf{x} \in \tilde{\Omega}$, of states in a system with finite link capacities are obtained by truncation

$$\tilde{\pi}(\mathbf{x}) = \mathrm{P}\{\mathbf{X} = \mathbf{x} \,|\, \mathbf{X} \in \tilde{\Omega}\} = \frac{\pi(\mathbf{x})}{\mathrm{P}\{\mathbf{X} \in \tilde{\Omega}\}}\,, \tag{7}$$

where $\mathrm{P}\{\mathbf{X} \in \tilde{\Omega}\} = \sum_{\mathbf{x} \in \tilde{\Omega}} \pi(\mathbf{x})$. This follows from the assumed detailed balance. See Kelly [12] for discussion of truncation.

2.3 Blocking

In a finite capacity network, blocking occurs whenever a user tries to establish a connection for channel i and layer r, and there is at least one link $j \in \mathcal{R}_u$ where the channel is on state $l < r$ and there is not enough spare capacity for setting the channel on the requested layer. Without loss of generality, the channels are ordered so that the blocking probability is calculated for channel with index I. Consider link j. A request for layer r succeeds if there is enough capacity already reserved for the layer in link j, or there is enough free capacity in the link, i.e. $\max\{d(r), d(y_{j,I})\} \le C_j - D'(\mathbf{y}_j)$. The expression "link j blocks" means that this condition does not hold for link j. The set $\mathcal{B}_{u,r}$ consists of the states where at least one link blocks for connection (u, I, r), when layer r of channel I is requested by user u, and is defined as

$$\mathcal{B}_{u,r} = \left\{ \mathbf{x} \in \tilde{\Omega} \,\middle|\, \exists j \in \mathcal{R}_u : d(r) > C_j - D'(\mathbf{y}_j) \right\}. \tag{8}$$

Then the time blocking probability for connection (u, I, r) is

$$B_{u,r} = \mathrm{P}\{\mathbf{X} \in \mathcal{B}_{u,r} \,|\, \mathbf{X} \in \tilde{\Omega}\} = \frac{\mathrm{P}\{\mathbf{X} \in \mathcal{B}_{u,r}\}}{\mathrm{P}\{\mathbf{X} \in \tilde{\Omega}\}}\,. \tag{9}$$

Call blocking probabilities for users depend on the chosen user model, as discussed in [11]. Calculation of time blocking probabilities for layers is easy, but very time consuming: the number of states in the state space is $(L+1)^{UI}$. The following section attacks this problem using the inverse convolution method.

3 Divide and Conquer

This section discusses efficient estimation of time blocking probabilities by applying the algorithm developed by Lassila and Virtamo [9]. As the form of the stationary distribution $\pi(\mathbf{x})$ is known, a natural choice for simulation is the Monte Carlo method. The main problem in the simulation is to quickly get a good estimate for $\mathrm{P}\{\mathbf{X} \in \mathcal{B}_{u,r}\}$, i.e., the numerator in Eq. (9), especially in the case when the blocking probability $B_{u,r}$ is small. Note that $B_{u,r}$ also depends on $\mathrm{P}\{\mathbf{X} \in \tilde{\Omega}\}$ given by the denominator of Eq. (9). This probability is usually close to unity and is easy to estimate using the standard Monte Carlo method. Therefore, the rest of this paper concentrates on efficient methods for estimating $\mathrm{P}\{\mathbf{X} \in \mathcal{B}_{u,r}\}$.

First, section 3.1 divides the task of estimating $P(\mathcal{B}_{u,r})$ into simpler sub-problems. Then, each of the sub-problems is solved using importance sampling, as is described in section 3.2.

3.1 Decomposition

In order to divide the task of estimating $P(\mathcal{B}_{u,r})$ to simpler sub-problems, $\mathcal{B}_{u,r}$ is partitioned into sets $\mathcal{E}^j_{u,r}$. $\mathcal{E}^j_{u,r}$ is defined as the set of points in $\mathcal{B}_{u,r}$ where link j blocks but none of the links closer to user u block,

$$\begin{aligned} \mathcal{E}^j_{u,r} = \mathcal{B}_{u,r} \cap \Bigg\{ \mathbf{x} \in \tilde{\Omega} \,\Bigg|\, & d(r) > C_j - D'(\mathbf{y}_j) \;\wedge \\ & d(r) \le C_{j'} - D'(\mathbf{y}_{j'}), \forall j' \in \mathcal{R}^j_u \Bigg\}, \end{aligned} \tag{10}$$

where $\mathcal{R}^j_u$ denotes the set of links on the path from u to j, including link u but not link j. The $\mathcal{E}^j_{u,r}$ form a partitioning of $\mathcal{B}_{u,r}$, i.e. $\mathcal{B}_{u,r} = \bigcup_{j\in\mathcal{R}_u} \mathcal{E}^j_{u,r}$, and $\mathcal{E}^j_{u,r} \cap \mathcal{E}^{j'}_{u,r} = \emptyset$, when $j \; / \!\!\neq j'$. From this it follows that

$$\mathrm{P}\{\mathbf{X} \in \mathcal{B}_{u,r}\} = \sum_{j\in\mathcal{R}_u} \mathrm{P}\{\mathbf{X} \in \mathcal{E}^j_{u,r}\}\,. \tag{11}$$

The probability $\mathrm{P}\{\mathbf{X} \in \mathcal{E}^j_{u,r}\}$ can be thought of as the blocking probability contribution due to link j. It should be noted, however, that blocking in the states where several links block can be arbitrarily attributed to any of the blocking links. I use the convention which attributes it to the blocking link closest to the user.

3.2 Conditioning of $\mathbf{P}\{\mathbf{X} \in \mathcal{E}^j_{u,r}\}$

Equation (11) decomposes estimation of $\mathrm{P}\{\mathbf{X} \in \mathcal{B}_{u,r}\}$ into independent sub-problems of estimating the $\mathrm{P}\{\mathbf{X} \in \mathcal{E}^j_{u,r}\}$. For these estimation tasks, I introduce the superset $\mathcal{D}^j_{u,r} \supset \mathcal{E}^j_{u,r}$,

$$\mathcal{D}^j_{u,r} = \left\{ \mathbf{x} \in \Omega \,\middle|\, d(r) > C_j - D'(\mathbf{y}_j) \geq d(y_{j,I}) \right\} . \tag{12}$$

This set corresponds to blocking states in a system where link j has a finite capacity C_j but all other links have infinite capacity. Since all links have finite capacity in real systems, and several links could block simultaneously, sets $\mathcal{D}^j_{u,r}$ are not disjoint unlike their subsets $\mathcal{E}^j_{u,r}$.

The next step is to use conditional probabilities to estimate $\mathrm{P}\{\mathbf{X} \in \mathcal{E}^j_{u,r}\}$, as follows:

$$\mathrm{P}\{\mathbf{X} \in \mathcal{E}^j_{u,r}\} = \mathrm{P}\{\mathbf{X} \in \mathcal{E}^j_{u,r} \,|\, \mathbf{X} \in \mathcal{D}^j_{u,r}\} \mathrm{P}\{\mathbf{X} \in \mathcal{D}^j_{u,r}\} . \tag{13}$$

This relation is useful from the simulation point of view since it is easy to compute $\mathrm{P}\{\mathbf{X} \in \mathcal{D}^j_{u,r}\}$ and to generate samples from the original distribution under the condition $\mathbf{X} \in \mathcal{D}^j_{u,r}$, as explained later. Monte Carlo simulation is then used to estimate the conditional probability $\mathrm{P}\{\mathbf{X} \in \mathcal{E}^j_{u,r} \,|\, \mathbf{X} \in \mathcal{D}^j_{u,r}\}$ instead of $\mathrm{P}\{\mathbf{X} \in \mathcal{E}^j_{u,r}\}$, which is usually much more effective.

Let $\widehat{\eta}^j_{u,r}$ denote the estimator for $\eta^j_{u,r} = \mathrm{P}\{\mathbf{X} \in \mathcal{E}^j_{u,r}\}$,

$$\widehat{\eta}^j_{u,r} = \frac{v_j}{N_j} \sum_{n=1}^{N_j} 1_{\mathbf{X}^*_n \in \mathcal{E}^j_{u,r}} , \tag{14}$$

where $v_j = \mathrm{P}\{\mathbf{X} \in \mathcal{D}^j_{u,r}\}$ and $\mathbf{X}^*_n$ denotes samples drawn from the conditional distribution $\mathrm{P}\{\mathbf{X} = \mathbf{x} | \mathbf{X} \in \mathcal{D}^j_{u,r}\}$. Then, the estimator for $P(\mathcal{B}^j_{u,r})$ is simply

$$\widehat{P}(\mathcal{B}^j_{u,r}) = \sum_{j \in \mathcal{R}_u} \widehat{\eta}^j_{u,r} . \tag{15}$$

Given the total number of samples N to be used for the estimator, the number of samples N_j allocated to each sub-problem is a free parameter. This can be exploited by assigning the number of samples to different $\widehat{\eta}^j_{u,r}$ according to their estimated variance during the simulation. See e.g. [8].

4 Inverse Convolution

This section presents the inverse convolution method (IC) for sample generation. I am now only considering the estimation of one $\eta^j_{u,r}$ for fixed $j \in \mathcal{R}_u$ and traffic class (u, I, r). The method is based on generating points from the conditional distribution $\mathrm{P}\{\mathbf{X} = \mathbf{x} \,|\, \mathbf{X} \in \mathcal{D}^j_{u,r}\}$ by reversing the steps used to calculate the

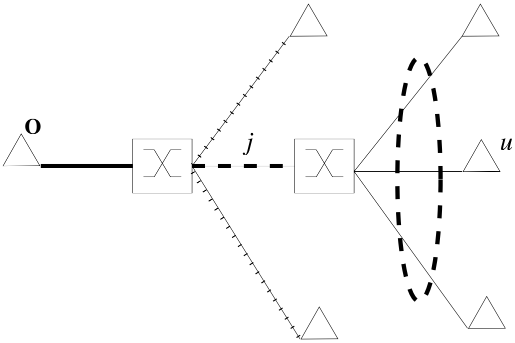


Fig. 1. Example of sample generation. A sample in the set $\mathcal{D}_{u,r}^{j}$ is generated for the link j (thick dashed line). States of the links marked by the dashed ellipse are generated by inverse convolution from the state of link j. States for links denoted by ticks are generated by a simple draw. The state of the link denoted by the thick line is calculated directly from the states of the other links.

occupancy distribution of the considered link. Note that the condition $\mathbf{X} \in \mathcal{D}_{u,r}^{j}$ is a condition expressed in terms of the occupancy, S_j', of the considered link. The idea in the inverse convolution method is to first generate a sample of $\mathbf{Y}_j$ such that the occupancy of the link is in the blocking region. Then, given the state $\mathbf{Y}_j$, the state of the network, i.e. states of the leaf links, is generated. The mapping $\mathbf{x} \mapsto \mathbf{y}_j$ is surjective, having several possible network states $\mathbf{x}$ generating the link state $\mathbf{y}_j$, and one of them is drawn according to their probabilities.

The main steps of the simulation can be summarized as follows (See Figure 1.):

1. Generate the states for leaf links u by
   a) Generate a sample state $\mathbf{Y}_j$ under the condition $d(r) > C_j - D'(\mathbf{y}_j) \geq d(y_{j,I})$ for link j.
   b) Generate the leaf link states $\mathbf{Y}_u$, $u \in \mathcal{U}_j$, with the condition that link j state $\mathbf{Y}_j = \max_{u \in \mathcal{U}_j}(\mathbf{Y}_u)$ is given.
   c) Generate the states $\mathbf{Y}_u$, $u \in \mathcal{U} - \mathcal{U}_j$ for the rest of the leaf links as in the normal Monte Carlo simulation.
2. The sample state of the network $\mathbf{X}_n^* \in \mathcal{D}_{u,r}^{j}$ consists of the set of all sample states of leaf links generated with step 1.
3. To collect the statistics for estimator $\widehat{\eta}_{u,r}^{j}$, check if $\mathbf{X}_n^* \in \mathcal{E}_u^j$.

The above steps are repeated for generating N_j samples. Section 4.1 explains the method of generating a sample for link j (step 1a). Section 4.2 explains the method for generating the leaf link states from the link state (step 1b).

4.1 Generating a Sample for $\mathcal{D}_{u,r}^{j}$

As already noted, I have partitioned the set of blocking states into disjoint sets $\mathcal{E}_{u,r}^{j}$. It is not easy to generate samples directly to these sets, however. Instead, I generate samples to sets $\mathcal{D}^j_{u,r}$ which correspond to the states in which at least link j blocks. After that it is possible to check if the sample belongs to the set $\mathcal{E}^j_{u,r}$ to collect the sum in Eq. (14).

Convolution method for calculating $\mathbf{P}\{\mathbf{X} \in \mathcal{D}^j_{u,r}\}$. First, the link occupancy S_j is easily calculated recursively as follows. Let $S_{j,i}$ denote link occupancy due to the first i channels,

$$S_{j,i} = \sum_{i' \leq i} d(Y_{j,i'}) \,. \tag{16}$$

Then $S_j = S_{j,I}$ and $S'_j = S_{j,I-1}$. The $Y_{j,i}$ are mutually independent, and $S_{j,i} = S_{j,i-1} + d(Y_{j,i})$, where $S_{j,i-1}$ and $Y_{j,i}$ are independent.

Channel I must be dealt with differently than the other channels, since the system can be in a blocking state only if $C_j - S_{j,I-1} < d(r)$, but the channel I can be in any state $l < r$. Knowing this, the set $\mathcal{D}^j_{u,r}$ can be partitioned into r point-wise disjoint subsets:

$$\begin{aligned}\mathcal{D}^{j,l}_{u,r} = \Big\{\mathbf{x} \in \Omega \,\Big|\, & y_{j,I} = l \,\wedge \\ & d(r) > C_j - D'(\mathbf{y}_j) \geq d(l)\Big\}, \qquad l \in \{0, \dots, r-1\} \,.\end{aligned} \tag{17}$$

If a state $\mathbf{x}$ belongs to the set $\mathcal{D}^{j,l}_{u,r}$, the state is a blocking state for link j, and the channel I is on layer l. Thus, the free capacity $C_j - D'(\mathbf{y}_j)$ of the link must be at most $d(r) - 1$, for the state to be a blocking state. The other channels may, however, consume at most $C_j - d(l)$ capacity units for the state to be within the allowed states. Now, let $v_j(l)$ denote the probability $\mathrm{P}\{\mathbf{X} \in \mathcal{D}^{j,l}_{u,r}\}$:

$$v_j(l) = p_{j,I}(l) \sum_{i=C_j-d(r)+1}^{C_j-d(l)} q_{j,I-1}(i) \,, \tag{18}$$

where $q_{j,i}(x) = \mathrm{P}\{S_{j,i} = x\}$. The probability mass v_j of the set $\mathcal{D}^j_{u,r}$, can be calculated as

$$v_j = \mathrm{P}\{\mathbf{X} \in \mathcal{D}^j_{u,r}\} = \sum_{l=0}^{r-1} v_j(l) \,. \tag{19}$$

The link occupancy distribution $q_{j,I-1}(\cdot)$ can be calculated recursively by convolution:

$$q_{j,i}(x) = \sum_{y=0}^{x} q_{j,i-1}(x - d(y)) p_{j,i}(y) \,, \tag{20}$$

where the recursion starts with $q_{j,0}(x) = 1_{x=0}$. Here, $p_{j,i}(y) = \mathrm{P}\{Y_{j,i} = y\}$, and is calculated easily, as shown in section 4.2.

Inverse convolution. For interpretation of the convolution step, note that the event $\{S_{j,i} = x\}$ is the union of the events $\{Y_{j,i} = y, S_{j,i-1} = x - d(y)\}$, $y \in \{0, \dots, L\}$. The corresponding probability is $q_{j,i-1}(x - d(y))p_{j,i}(y)$. Conversely, the conditional probability of the event $\{Y_{j,i} = y,\, S_{j,i-1} = x - d(y)\}$ given that $S_{j,i} = x$ is,

$$\mathrm{P}\{Y_{j,i} = y,\, S_{j,i-1} = x - d(y) \,|\, S_{j,i} = x\} = \frac{p_{j,i}(y)q_{j,i-1}(x - d(y))}{q_{j,i}(x)}\,. \tag{21}$$

Generating a sample state in $\mathcal{D}^j_{u,r}$ starts by drawing a value l for $Y_{j,I}$ using the distribution

$$\mathrm{P}\{Y_{j,I} = l \,|\, \mathbf{X} \in \mathcal{D}^j_{u,r}\} = \frac{\mathrm{P}\{Y_{j,I} = l,\, \mathbf{X} \in \mathcal{D}^j_{u,r}\}}{\mathrm{P}\{\mathbf{X} \in \mathcal{D}^j_{u,r}\}} = \frac{v_j(l)}{v_j}\,, \tag{22}$$

where $l \in \{0, \dots, r-1\}$.

Then, a value for $S'_j = S_{j,I-1}$ is drawn with the condition that $Y_{j,I} = l$ that is, using the distribution

$$p(x|l) = \mathrm{P}\{S_{j,I-1} = x \,|\, Y_{j,I} = l,\, \mathbf{X} \in \mathcal{D}^j_{u,r}\} = \frac{\mathrm{P}\{Y_{j,I} = l, S_{j,I-1} = x\}}{\mathrm{P}\{Y_{j,I} = l,\, \mathbf{X} \in \mathcal{D}^j_{u,r}\}}\,, \tag{23}$$

since $\{Y_{j,I} = l \wedge S_{j,I-1} = x\} \Rightarrow \{\mathbf{X} \in \mathcal{D}^j_{u,r}\}$, restricting x to $x \in \{C_j - d(r) + 1, \dots, C_j - d(l)\}$, and

$$p(x|l) = \frac{p_{j,I}(l)q_{j,I-1}(x)}{v_j(l)} = \frac{q_{j,I-1}(x)}{\sum_{y=C_j-d(r)+1}^{C_j-d(l)} q_{j,I-1}(y)}\,. \tag{24}$$

Then, given the value of $S_{j,I-1}$, the state $Y_{j,i}$ of each channel $(i = I-1, \dots, 1)$ is drawn in turn using probabilities in Eq. (21). Concurrently with the state $Y_{j,i}$, the value of $S_{j,i-1}$ becomes determined. This is then used as the conditioning value in the next step to draw the value of $Y_{j,i-1}$ (and of $S_{j,i-2}$), etc. Note that for reasonable sizes of links, it is advantageous to store the probabilities for fast generation of samples.

The next subsection presents a method for drawing leaf link states $\mathbf{Y}_u$, given the state $\mathbf{Y}_j$ of link j.

4.2 Generating Leaf Link States from a Link State

Having drawn a value for state $\mathbf{Y}_j$ of link j, it is possible to draw values of the state vectors $\mathbf{Y}_u$, $u \in \mathcal{U}$, of the leaf links. For $u \in \mathcal{U}_j$, states $\mathbf{Y}_u$ are generated under the condition $\mathbf{Y}_j = \max_{u \in \mathcal{U}_j}(\mathbf{Y}_u)$ using a similar inverse convolution procedure as above. Due to the assumed independence of channels, this condition can be broken down into separate conditions, i.e. for each i there is a separate problem of generating the values $Y_{u,i}$, $u \in \mathcal{U}$, under the condition $Y_{j,i} = \max_{u \in \mathcal{U}_j}(Y_{u,i})$ with a given $Y_{j,i}$. The above conditions affect leaf links

$u \in \mathcal{U}_j$. For other links $u \in \mathcal{U} - \mathcal{U}_j$, the states $\mathbf{Y}_u$ are independently generated from the distribution $\pi_u(\cdot)$.

First, let us consider a convolutional approach for generating a link state for channel i and link j if the states for each link $u \in \mathcal{U}_j$ are already known. In this section, I use an index $u_j \in \{1, \ldots, U_j\} = \mathcal{U}_j$ for the subset of leaf links. Let $Z_{u_j,i} = x$ denote the event that the channel i is on state x on link j when $u' = 1, \ldots, u_j$ leaf links have been counted for, i.e. $Z_{u_j,i} = \max_{u' \le u_j}(Y_{u',i})$. Note that $Y_{j,i} = Z_{U_j,i}$. Probabilities $\xi_{u_j,i}(s) = \mathrm{P}\{Z_{u_j,i} = s\}$ can be calculated recursively as follows:

$$\xi_{u_j,i}(s) = p_{u_j,i}(s) \sum_{s'=0}^{s-1} \xi_{u_j-1,i}(s') + \xi_{u_j-1,i}(s) \sum_{s'=0}^{s} p_{u_j,i}(s') \,. \tag{25}$$

The recursion starts with $\xi_{0,i}(s) = 1_{s=0}$. The probabilities $p_{j,i}(s)$ used in the previous section are then simply $p_{j,i}(s) = \xi_{U_j,i}(s)$ where all users have been taken into account. If $Z_{u_j-1,i} = s$, then necessarily $Z_{u_j,i} \ge s$ (due to the nature of max-operation).

Conversely, to generate the state for each leaf link, given the value of $Y_{j,i}$, I first generate $Z_{u_{j-1},i}$ from the distribution:

$$\mathrm{P}\{Z_{u_j-1,i} = x \,|\, Z_{u_j,i} = s\} = \begin{cases} \dfrac{\xi_{u_j-1,i}(x) \sum_{s'=0}^{x} p_{u_j,i}(s')}{\xi_{u_j,i}(s)}, & \text{when } x = s\,, \\ \dfrac{\xi_{u_j-1,i}(x) p_{u_j,i}(s)}{\xi_{u_j,i}(s)}, & \text{otherwise}\,. \end{cases} \tag{26}$$

Note that the event $Z_{u_j-1,i} < Z_{u_j,i}$ implies directly that $Y_{u_j,i} = Z_{u_j,i}$. If this is not the case, the value of $Y_{u_j,i}$ is drawn from the distribution

$$\mathrm{P}\{Y_{u_j,i} = y \,|\, Z_{u_j-1,i} = Z_{u_j,i} = s\} = \frac{p_{u_j,i}(y)}{\sum_{y'=0}^{s} p_{u_j,i}(y')} \,. \tag{27}$$

This procedure is repeated for each channel. The state vectors of each leaf link $u \in \mathcal{U}_j$ result from this procedure. The rest of the leaf link states must be generated as in the normal Monte Carlo simulation using distribution $\pi_u(\cdot)$.

5 Numerical Results

This section gives some numerical examples to illustrate the efficiency of the presented method in Monte Carlo simulation of the blocking probabilities. I consider the same network used in [7]. The network is the one shown in Figure 1. There is a root node, four channels, $I = 4$, and three layers, $L = 3$, with $d(l) = l$ for all channels. The capacity of the root link is $C_J = 6$, for the others, $C_j = 5$. Each leaf link has an infinite user population offering traffic to each channel. The probability $p_{u,i}(l)$ that a channel is on layer l is $p_{u,i}(l) = \alpha_l b$ (for all users), where $\alpha_1 = 0.3$, $\alpha_2 = 0.2$ and $\alpha_3 = 0.1$. I simulated blocking for channel I and

Table 1. The relative deviation of the estimates $\widehat{P}(\mathcal{B}_{u,r})$ for the example network

Samples	b	r / $B_{u,r}$	relative deviation		
			MC	MC-IC	MC-ICSA
10 000	0.01	1 / 0.0146%	0.6301	0.0060	0.0048
		2 / 0.0591%	0.4244	0.0063	0.0049
		3 / 0.98%	0.0948	0.0078	0.0049
	0.05	1 / 0.33%	0.1240	0.0067	0.0056
		2 / 1.28%	0.0748	0.0068	0.0055
		3 / 6.12%	0.0384	0.0076	0.0060
	0.10	1 / 1.14%	0.0564	0.0073	0.0063
		2 / 4.25%	0.0413	0.0073	0.0060
		3 / 14.0%	0.0227	0.0079	0.0068
100 000	0.01	1 / 0.0146%	0.2075	0.0019	0.0015
		2 / 0.0591%	0.1273	0.0020	0.0015
		3 / 0.98%	0.0281	0.0025	0.0016
	0.05	1 / 0.33%	0.0398	0.0021	0.0018
		2 / 1.28%	0.0227	0.0022	0.0017
		3 / 6.12%	0.0128	0.0024	0.0019
	0.10	1 / 1.14%	0.0194	0.0023	0.0020
		2 / 4.25%	0.0120	0.0023	0.0019
		3 / 14.0%	0.0073	0.0025	0.0021

user u (the longer path) with three values for b: 0.01, 0.05, and 0.1 to compare the simulation methods in light, moderate, and high load conditions.

I also estimated the relative deviation of the estimator for 10^4 samples and 10^5 samples, given by $(\mathrm{V}[\widehat{P}(\mathcal{B}_{u,r})])^{1/2}/\widehat{P}(\mathcal{B}_{u,r})$. For classic Monte Carlo (MC), these were the total numbers of samples used, while for Inverse Convolution method (MC-IC), one third of samples was used for each estimate $\widehat{\eta}^j_{u,r}$. For Inverse Convolution with optimal Sample Allocation (MC-ICSA) [8], the total number of samples was allocated optimally for each estimate.

The results are shown in Table 1. The table shows that the variance reductions obtained with the inverse convolution method are remarkable. For example, for light load ($b = 0.01$), the ratio between the deviations of the standard MC and the inverse convolution method (MC-ICSA) is up to 131 for 10 000 samples and 138 for 100 000 samples, corresponding to a decrease by a factor of 17 000 to 19 000 in the required sample sizes. In high load situations, the overhead in sample generation might not be justified, as the traditional Monte Carlo method gives rather good estimates, too.

6 Summary

I presented an algorithm for efficient simulation of time blocking probabilities for multi-layer multicast streams with the assumption that blocked calls are lost.

Calculating blocking probabilities for this system directly from the steady state probabilities is easy in principle, but excessively time-consuming.

The simulation algorithm presented is based on the inverse convolution algorithm. The results in the shown example network support convincingly its efficiency, yielding a decrease in sample size of up to a factor of 19 000 over the traditional Monte Carlo method.

Acknowledgement. This study was funded by the Academy of Finland, and also supported by Nokia Foundation. I thank Pasi Lassila, Jorma Virtamo and Samuli Aalto for their helpful advise.

References

1. Fortet R. and Grandjean C., "Congestion in a loss system when some calls want several devices simultaneously," *Electrical Communication*, vol. 39, no. 4, pp. 513–526, 1964.
2. Ross K. W., *Multiservice Loss Models for Broadband Telecommunication Networks*, Springer Verlag, London, 1995.
3. Diot C., Dabbous W., and Crowcroft J., "Multipoint communication: A survey of protocols, functions, and mechanisms," *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 3, pp. 277–290, Apr. 1997.
4. Karlsson G. and Vetterli M., "Packet video and its integration into the network architecture," *IEEE Journal on Selected Areas in Communications*, vol. 7, no. 5, pp. 739–751, June 1989.
5. Chan W. C. and Geraniotis E., "Tradeoff between blocking and dropping in multicasting networks," in *ICC '96 Conference Record*, June 1996, vol. 2, pp. 1030–1034.
6. Karvo J., Virtamo J., Aalto S., and Martikainen O., "Blocking of dynamic multicast connections," *Telecommunication Systems*, vol. 16, no. 3,4, pp. 467–481, 2001.
7. Nyberg E., Virtamo J., and Aalto S., "An exact algorithm for calculating blocking probabilities in multicast networks," in *Networking 2000*, Pujolle G., Perros H., Fdida S., Körner U., and Stavrakakis I., Eds., Paris, May 2000, pp. 275–286.
8. Lassila P., Karvo J., and Virtamo J., "Efficient importance sampling for Monte Carlo simulation of multicast networks," in *Proc. INFOCOM'01*, Anchorage, Alaska, Apr. 2001, pp. 432–439.
9. Lassila P. E. and Virtamo J. T., "Nearly optimal importance sampling for Monte Carlo simulation of loss systems," *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, vol. 10, no. 4, pp. 326–347, Oct. 2000.
10. Karvo J., Aalto S., and Virtamo J., "Blocking probabilities of two-layer statistically indistinguishable multicast streams," in *Proc. International Teletraffic Congress ITC-17*, de Souza J. M., Fonseca N. L. S., and de Souza e Silva E. A., Eds., Salvador da Bahia, Brazil, Sept. 2001, pp. 769–779.
11. Karvo J., Aalto S., and Virtamo J., "Blocking probabilities of multi-layer multicast streams," in *2002 Workshop on High Performance Switching and Routing (HPSR 2002) (To appear)*, Kobe, Japan, May 2002.
12. Kelly F. P., *Reversibility and Stochastic Networks*, John Wiley & Sons, 1979.

Aggregated Multicast – A Comparative Study*

Jun-Hong Cui, Jinkyu Kim, Dario Maggiorini, Khaled Boussetta, and Mario Gerla

Computer Science Department, University of California, Los Angeles, CA 90095

Abstract. Multicast state scalability is among the critical issues which delay the deployment of IP multicast. In our previous work, we proposed a scheme, called aggregated multicast to reduce multicast state. The key idea is that multiple groups are forced to share a single delivery tree. We presented some initial results to show that multicast state can be reduced. In this paper, we develop a more quantitative assessment of the cost/benefit trade-offs. We introduce metrics to measure multicast state and tree management overhead for multicast schemes. We then compare aggregated multicast with conventional multicast schemes, such as source specific tree scheme and shared tree scheme. Our extensive simulations show that aggregated multicast can achieve significant routing state and tree management overhead reduction while containing the expense of extra resources (bandwidth waste and tunnelling overhead, etc.). We conclude that aggregated multicast is a very cost-effective and promising direction for scalable transit domain multicast provisioning.

1 Introduction

IP Multicast has been a very hot area of research, development and testing for more than one decade since Stephen Deering established the IP multicast model in 1988 [6]. However, IP multicast is still far from being widely deployed in the Internet. Among the issues which delay the deployment, state scalability is one of the most critical ones.

IP multicast utilizes a tree delivery structure on which data packets are duplicated only at fork nodes and are forwarded only once over each link. By doing so IP multicast can scale well to support very large multicast groups. However, a tree delivery structure requires all tree nodes to maintain per-group (or even per-group/source) forwarding information, which increases linearly with the number of groups. Growing number of forwarding state entries means more memory requirement and slower forwarding process since every packet forwarding action involves an address look-up. Thus, multicast scales well to the number of members within a single multicast group. But, it suffers from scalability problems when the number of simultaneous active multicast groups is very large.

To improve multicast state scalability, we proposed a novel scheme to reduce multicast state, which we call *aggregated multicast*. In this scheme, multiple multicast groups are forced to share one distribution tree, which we call an *aggregated tree*. This way, the number of trees in the network may be significantly reduced. Consequently, forwarding state is also reduced: core routers only need to keep state per aggregated tree instead

* This material is based upon work supported by the National Science Foundation under Grant No. 9805436, and CISCO/CORE fund No. 99-10060

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 1032–1044, 2002.

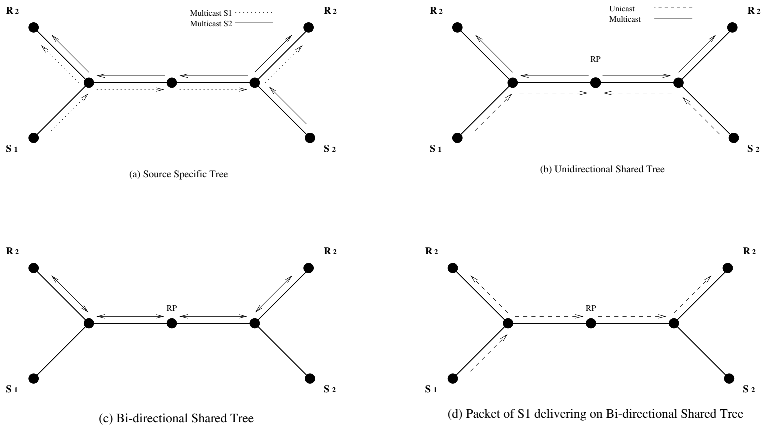
of per group. The trade-off is that this approach may waste extra bandwidth to deliver multicast data to non-group-member nodes. In our earlier work [8,9], we introduced the basic concept of aggregated multicast, proposed an algorithm to assign multicast groups to delivery trees with controllable bandwidth overhead and presented some initial results to show that multicast state can be reduced through inter-group tree sharing. However, a thorough performance evaluation of aggregated multicast is needed: what level of the gain does aggregated multicast offer over conventional multicast schemes? In this paper, we propose metrics to measure multicast state and tree management overhead for multicast schemes. We then compare aggregated multicast with conventional multicast schemes, such as source specific tree scheme and shared tree scheme. Our extensive simulations show that aggregated multicast can achieve significant state and tree management overhead reduction while at reasonable expense (bandwidth waste and tunnelling overhead, etc.).

The rest of this paper is organized as follows. Section 2 gives a classification of multicast schemes. Section 3 reviews the concept of aggregated multicast and presents a new algorithm for group-tree matching. Section 4 then discusses the implementation issues for different multicast schemes and defines metrics to measure multicast state and tree management overhead, and Section 5 provides an extensive simulation study of different multicast schemes. Finally Section 6 summarizes the contributions of our work.

2 A Classification of Multicast Schemes

According to the type of delivery tree, we classify the existing intra-domain multicast routing protocols into two categories (It should be noted that, in this paper, we only consider intra-domain multicasting): in the first category, protocols construct source specific tree, and in the second category, protocols utilize shared tree. For the convenience of discussion, we call the former category as **source specific tree scheme**, and the latter one as **shared tree scheme**. According to this classification, we can say, DVMRP [12], PIM-DM [5], and MOSPF [11] belong to source specific tree scheme category, while CBT [3], PIM-SM [7], and BIDIR-PIM [10] are basically shared tree schemes (of course, PIM-SM can also activate source specific tree when needed).

Source specific tree scheme constructs a separate delivery tree for each source. Namely, each source of a group utilizes its own tree to deliver data to the receivers in the group. The shared tree scheme instead constructs trees based on per-group and all the sources of a group use the same tree to deliver data to the receivers. In other words, multiple sources of the same group share a single delivery tree. Shared tree can be unidirectional or bi-directional. PIM-SM is a unidirectional shared tree scheme. CBT and BIDIR-PIM are bi-directional shared tree schemes. Fig. 1 shows the different types of trees for the same group G with sources $(S1, S2)$ and receivers $(R1, R2)$. For source specific tree schemes, two trees are set up for group G. For the unidirectional shared tree scheme, one tree is set up. Each source needs to unicast packets to the rendezvous point (RP) or build source specific state on all nodes along the path between the source and the RP. For the last scheme, only one bi-directional tree will work. A source can



(a) Source Specific Tree

(b) Unidirectional Shared Tree

(c) Bi-directional Shared Tree

(d) Packet of S1 delivering on Bi-directional Shared Tree

Fig. 1. Different types of trees for group G with sources $(S1, S2)$ and receivers $(R1, R2)$.

unicast packet to the nearest on-tree node instead of RP. And each on-tree node can deliver packets along the bi-directional tree.

Compared with conventional multicast schemes, aggregated multicast raises tree-sharing to an even higher level—inter-group tree sharing, where multiple multicast groups are forced to share one aggregated tree. An aggregated tree can be either a source specific tree or a shared tree, while a shared tree can be either unidirectional or bi-directional. We are going to review the basic concept of aggregated multicast and discuss some related issues in the following section.

3 Aggregated Multicast

3.1 Concept of Aggregated Multicast

Aggregated multicast [8,9] is proposed to reduce multicast state, and it is targeted to intra-domain multicast provisioning. The key idea is that, instead of constructing a tree for each individual multicast group in the core network (backbone), multiple multicast groups are forced to share a single aggregated tree.

Fig. 2 illustrates a hierarchical inter-domain network peering. Domain A is a regional or national ISP's backbone network, and domain D, X, and Y are customer networks of

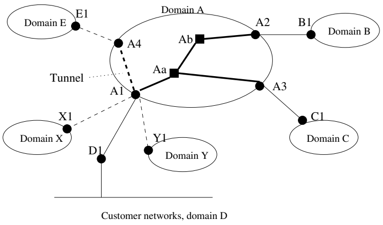


Fig. 2. Domain peering and a cross-domain multicast tree, tree nodes: D1, A1, Aa, Ab, A2, B1, A3, C1, covering group G_0 (D1, B1, C1).

domain A at a certain location (say, Los Angeles), and domain E is a customer network of domain A in another location (say, Seattle). Domain B and C can be other customer networks (say, in Boston) or some other ISP's networks that peer with A. A multicast session originates at domain D and has members in domain B and C. Routers D1, A1, A2, A3, B1 and C1 form the multicast tree at the inter-domain level while A1, A2, A3, Aa and Ab form an intra-domain sub-tree within domain A (there may be other routers involved in domain B and C). Consider a second multicast session that originates at domain D and also has members in domain B and C. For this session, a sub-tree with exactly the same set of nodes will be established to carry its traffic within domain A. Now if there is a third multicast session that originates at domain X and it also has members in domain B and C, then router X1 instead of D1 will be involved, but the sub-tree within domain A still involves the same set of nodes: A1, A2, A3, Aa, and Ab.

To facilitate our discussions, we make the following definitions. For a group G, we call **terminal nodes** the nodes where traffic enters or leaves a domain, A1, A2, and A3 in our example. We call **transit nodes** the tree nodes that are internal to the domain, such as Aa and Ab in our example.

In conventional IP multicast, all the nodes in the above example that are involved within domain A must maintain separate state for each of the three groups individually though their multicast trees are actually of the same "shape". Alternatively, in the aggregated multicast, we can setup a pre-defined tree (or establish a tree on demand) that covers nodes A1, A2 and A3 using a single multicast group address (within domain A). This tree is called an **aggregated tree** (AT) and it is shared by more than one multicast groups (three groups in the above example). We say an aggregated tree T **covers** a group G if all terminal nodes for G are member nodes of T. Data from a specific group is encapsulated at the incoming terminal node using the address of the aggregated tree. It is then distributed over the aggregated tree and decapsulated at exiting terminal nodes to be further distributed to neighboring networks. This way, transit router Aa and Ab only need to maintain a single forwarding entry for the aggregated tree regardless how many groups are sharing it.

Thus, aggregated multicast can reduce the required multicast state. Transit nodes don't need to maintain state for individual groups; instead, they only maintain forwarding state for a smaller number of aggregated trees. The management overhead for the distribution trees is also reduced. First, there are fewer trees that exchange refresh messages. Second, tree maintenance can be a much less frequent process than in conventional multicast, since an aggregated tree has a longer life span.

3.2 Group-Tree Matching in Aggregated Multicast

Aggregated multicast achieves state reduction through inter-group tree sharing—multiple groups share a single aggregated tree. When a group is started, an aggregated tree should be assigned to the group following some rules. If a dense set of aggregated trees is pre-defined, things will be easy: just choose the tree with minimum cost which can cover the group. While in the dynamic case (aggregated tree are established on demand), a more elaborate group-tree matching algorithm is needed.

When we try to match a group G to an aggregated tree T, we have four cases:

1. T can cover G and all the tree leaves are terminal nodes for G, then this match is called **perfect match** for G;
2. T can cover G but some of the tree leaves are not terminal nodes for G, then this match is a **pure-leaky match** (for G);
3. T can not cover G and all the tree leaves are terminal nodes for G, then this match is called a **pure-incomplete match**;
4. T can not cover G and some of the tree leaves are not terminal nodes for G, we name this match as **incomplete leaky match**.

Namely, we denote the case when some of the tree leaves are not terminal nodes for the group G as **leaky match** and the case when the tree can not cover the group G as **incomplete match**. Clearly, leaky match includes case 2 and 4, and incomplete match includes case 3 and 4.

To give examples, the aggregated tree T_0 with nodes (A1, A2, A3, Aa, Ab) in Fig. 2 is a perfect match for our early multicast group G_0 which has members (D1, B1, C1). However, if the above aggregated tree T_0 is also used for group G_1 which only involves member nodes (D1, B1), then it is a pure-leaky match since traffic for G_1 will be delivered to node A3 (and will be discarded there since A3 does not have state for that group). Obviously, the aggregated tree T_0 is an pure-incomplete match for multicast group G_2 which has members (D1, B1, C1, E1) and an incomplete leaky match for multicast group G_3 with members (D1, B1, E1).

We can see that leaky match helps to improve inter-group tree sharing. A disadvantage of leaky match is that some bandwidth is wasted to deliver data to nodes that are not members for the group. Leaky match may be unavoidable since usually it is not possible to establish aggregated trees for all possible group combinations. In the incomplete match case, we have two ways to get a tree for the group. One way is to construct a bigger tree by moving the entire group to a new larger aggregated tree, or, to extend the current aggregated tree to a bigger tree. Extending a tree might involve a lot of overhead, because all the groups which use the extended aggregated tree need to make the corresponding adjustment. An alternative way is to use "tunnelling". Here we give an example. Suppose member E1 in domain E decides to join group G_0 in Fig. 2. Instead of constructing a bigger tree, an extension "tunnel" can be established between edge router A4 (connecting domains A and E) and edge router A1. This solution combines features of multicast inter-group tree sharing and tunnelling; it still preserves core router scalability properties by pushing complexity to edge routers. We can see that, if we employ tunnelling instead of tree extension, then an incomplete match only involves tunnelling. An incomplete leaky match will activate tunnelling and will also waste resources because of leaky matching.

3.3 A New Group-Tree Matching Algorithm

Here we present a new group-tree matching algorithm which is used in our simulation. To avoid the overhead of tree extension, this algorithm uses tunnelling for incomplete match. First, we introduce some notations and definitions.

Overhead Definition. A network is modelled as an undirected graph $G(V, E)$. Each edge (i, j) is assigned a positive cost $c_{ij} = c_{ji}$, which represents the cost to transport a unit of data from node i to node j (or from j to i). Given a multicast tree T, total cost to distribute a unit of data over that tree is

$$C(T) = \sum_{(i,j)\in T} c_{ij}. \tag{1}$$

If every link is assumed to have equal cost 1, tree cost is simply $C(T) = |T| - 1$, where $|T|$ denotes the number of nodes in T. This assumption holds in this paper. Let MTS (Multicast Tree Set) denote the current set of multicast trees established in the network. A "native" multicast tree (constructed by some conventional multicast routing algorithm, denoted by A) for a multicast group G is denoted by T_G^A.

For any aggregated tree T, as mentioned in Section 3.2, it is possible that T does not have a perfect match with group G, which means that the match is leaky match or incomplete match. In leaky match case, some of the leaf nodes of T are not the terminal nodes for G, and then packets reach some destinations that are not interested in receiving them. Thus, there is bandwidth overhead in aggregated multicast. We assume each link has the same bandwidth, and each multicast group has the same bandwidth requirement, then it is easy to get that the percentage bandwidth overhead (denoted by $\delta_L(G,T)$) is actually equal to the percentage link cost overhead:

$$\delta_L(G,T) = \frac{C(T) - C(T_G^A))}{C(T_G^A)}, \tag{2}$$

Apparently, $\delta_L(G,T)$ is 0 for perfect match and pure-incomplete match.

In incomplete match case, T can not cover all the members of group G, and some tunnels need to be set up. Data packets of G exits from the leaf nodes of T, and tunnels to the corresponding terminal nodes of G. Clearly, there is tunnelling overhead caused by unicasting data packets to group terminal nodes. Each tunnel's cost can be measured by the link cost along the tunnel. Assume there are k_G tunnels for group G, and each tunnel is denoted by $T_{G,i}^t$, where $1 \leq i \leq k_G$, then we define the percentage tunnelling overhead for this incomplete match as

$$\delta_I(G,T) = \frac{\sum_{i=1}^{k_G} C(T_{G,i}^t)}{C(T_G^A)}. \tag{3}$$

It is easy to tell that $\delta_I(G,T)$ is 0 for perfect match and pure-leaky match.

Algorithm Description. Our new group-tree matching algorithm is based on bandwidth overhead and tunnelling overhead. Let l_t be the given bandwidth overhead threshold for leaky match, and t_t be the given tunnelling overhead threshold for incomplete match. When a new group is started,

1. compute a "native" multicast tree T_G^A for G based on the multicast group membership;
2. for each tree T in MTS, compute $\delta_L(G,T)$ and $\delta_I(G,T)$; if $\delta_L(G,T) < l_t$ and $\delta_I(G,T) < t_t$ then T is considered to be a candidate aggregated tree for G;

3. among all candidates, choose the one such that $f(\delta_L(G,T), \delta_I(G,T))$ is minimum and denote it as T_m, then T_m is used to deliver data for G; if T_m can not cover G, the corresponding tunnels will be set up;
4. if no candidate found in step 2, T_G^A is used for G and is added to MTS.

In step 3, $f(\delta_L(G,T), \delta_I(G,T))$ is a function to decide how to choose the final tree from a set of candidates. In our simulations,

$$f(\delta_L(G,T), \delta_I(G,T)) = \delta_L(G,T) + \delta_I(G,T). \tag{4}$$

Actually, this function can be chosen according to the need in the real scenarios. For example, we can give more weight to bandwidth overhead if bandwidth is our main concern.

4 Experiment Methodology

In an aggregated multicast scheme, sharing a multicast tree among multiple groups may significantly reduce the states at network core routers and correspondingly the tree management overhead. However, what level of gain can aggregated multicast get over other multicast schemes? In this section, we will discuss some implementation issues for different multicast schemes in our simulations, and define metrics to measure multicast state and tree management overhead. Then in Section 5, we will compare aggregated multicast with other multicast schemes through simulations.

4.1 Implementation of Multicast Schemes in SENSE

We do our simulations using SENSE (**S**imulation **E**nvironment for **N**etwork **S**ystem **E**volution) [2], which is a network simulator developed at the network research laboratory at UCLA to perform wired network simulation experiments.

In SENSE, we can support the source specific tree scheme, the shared tree scheme (with unidirectional tree and bi-directional tree), and the aggregated multicast scheme (with source specific tree, unidirectional shared tree and bi-directional shared tree). It should be noted that, the multicast schemes we discuss here are not specific multicast routing protocols, since the goal of this paper is to study the gain of aggregated multicast over conventional multicast schemes. The comparison is between schemes, not protocols.

We implement each multicast scheme with a centralized method. For each scheme, there is a centralized processing entity (called *multicast controller*), which has the knowledge of network topology and multicast group membership. The multicast controller is responsible for constructing the multicast tree according to different multicast schemes and then distributing the routing tables to the corresponding nodes. In the implementation, we did not model the membership acquisition and management procedures which depend on the specific multicast routing protocol. This omission reduces the bias and improves the fairness in comparing different multicast schemes. The multicast controller will read group and member dynamics from a pre-generated (or generated on-the-fly) trace file.

For shared tree scheme (either unidirectional or bi-directional) and aggregated multicast scheme with shared tree (unidirectional or bi-directional), a core node or a rendezvous point (RP) is needed when a tree is constructed. To achieve better load balancing, the core node should be chosen carefully. In our implementation, for all multicast schemes using shared trees, a set of possible core routers are pre-configured. Then, when a group is initialized, the core is chosen so as to minimize the cost of the tree.

In an aggregated multicast scheme, the multicast controller also needs to manage aggregated trees and multicast groups and manipulate group-tree matching algorithm. The multicast controller has the same responsibility as the tree manager (mentioned in [8,9]) in aggregated multicast. It collects group join messages and assigns aggregated trees to groups. Once it determines which aggregated tree to use for a group, the tree manager can install corresponding state at the terminal nodes involved.

4.2 Performance Metrics

The main purpose of tree sharing is to reduce multicast state and tree maintenance overhead. So, multicast state and tree management overhead measures are of most concern here. In our experiments, we introduce the following metrics.

Number of multicast trees (or **number of trees** for shorthand) is defined as $|MTS|$, where *MTS* denotes the current set of multicast trees established in the networks. This metric is a direct measurement for the multicast tree maintenance overhead. The more multicast trees, the more memory required and the more processing overhead involved (though the tree maintenance overhead depends on the specific multicast routing protocols).

Forwarding state in transit nodes (or **transit state** for shorthand). Without losing generality, we assume a router needs one state entry per multicast address in its forwarding table. As we defined in Section 3, in a multicast tree, there are transit nodes and terminal nodes. We note that forwarding state in terminal nodes can not be reduced in any multicast scheme. Even in aggregated multicast, the terminal nodes need to maintain the state information for individual groups. So, to assess the state reduction, we measure the forwarding state in transit nodes only.

5 Simulations

In this section, we compare aggregated multicast with conventional multicast schemes through extensive simulation, and quantitatively evaluate the gain of aggregated multicast.

5.1 Multicast Trace Generation

Multicast Group Models. Given the lack of experimental large scale multicast traces, we have chosen to develop membership models that exhibit locality and group correlation preferences. In our simulation, we use the group model previously developed in [9]: **The random node-weighted model**. For completeness, we provide here a summary description of this model.

The random node-weighted model. This model statistically controls the number of groups a node will participate in based on its weight: for two nodes i and j with weight $w(i)$ and $w(j)$ ($0 < w(i), w(j) \leq 1$), let $N(i)$ be the number of groups that have i as a member and $N(j)$ be the number of groups that have j as a member, then it is easy to prove that, in average, $\frac{N(i)}{N(j)} = \frac{w(i)}{w(j)}$. Assuming the number of nodes in the network is N and nodes are numbered from 1 to N. To each node i, $1 \leq i \leq N$, is assigned a weight $w(i)$, $0 \leq w(i) \leq 1$. Then a group can be generated as the following procedure:

```
for $i = 1$ to $N$ do
    generate p, a random number uniformly between 0 and 1, let it be p
    if $p < w(i)$ then
        add i as a group member
    end if
end for
```

Following this model, the average size of multicast groups is $N \sum_{i=1}^{n} w(i)$.

Multicast Membership Dynamics. Generally, there are two methods to control multicast group member dynamics. The first one is to create new members (sources and receivers) for a group according to some pre-defined statistics (arrive rate and member life time etc.), then decide the termination of a group based on the distribution of the group size. This is actually a member-driven dynamic. As to the other method, we call it group-driven dynamics, which means that, group characteristics (group size, group arrival rate, and group life time) are defined first and then group members are generated according to groups. In our experiment, we use the second method, in which the group statistics are controlled first (using the random node weighted model). Actually, the second method looks more reasonable for many real life multicast applications (such as video conference, tele-education, etc.). In any event, the specific method used to control group member dynamics is not expected to affect our simulation results.

In our experiment, given a group life period $[t_1, t_2]$, and the group member set g, where $|g| = n$, for any node $m_i \in g$, $1 \leq i \leq n$, its join time and leave time are denoted by $t_{join}(m_i)$ and $t_{leave}(m_i)$ separately. Then the member dynamics is controlled as follows:

```
for $i = 1$ to $n$ do
    $m_i \in g$
    $t_{join}(m_i)$=get_rand($t_1, t_2$); (get a random time in [$t_1, t_2$])
    $t_{leave}(m_i)$=get_rand($t_{join}(m_i), t_2$); (get a random time in [$t_{join}(m_i), t_2$])
end for
```

It is not difficult to know that the average life time of each member is $|t_2 - t_1|/4$.

5.2 Results and Analysis

We now present results from simulation experiments using a real network topology, vBNS backbone [1].

In vBNS backbone, there are 43 nodes, among which FORE ASX-1000 nodes (16 of them) are assumed to be *core routers* only (i.e. will not be terminal nodes for any multicast group) and are assigned weight 0. Any other node is assigned a weight 0.05 to 0.8 according to link bandwidth of the original backbone router – the rationale is that, the more the bandwidth on the outgoing (and incoming) links of a node, the more the number of multicast groups it may participate in. So, we assign weight 0.8 to nodes with OC-12C links (OC-12C-linked nodes for shorthand), 0.2 to nodes with OC-3C links (OC-3C-linked nodes), and 0.05 to nodes with DS-3 links (DS-3-linked nodes).

In simulation experiments, multicast session requests arrive as a Poisson process with arrival rate λ. Sessions' life time has an exponential distribution with average μ^{-1}. At steady state, the average number of sessions is $\bar{N} = \lambda/\mu$. During the life time of each multicast session, group members are generated dynamically according to group-driven method introduced earlier. Group membership is controlled using the random node-weighted model. Performance data is collected at certain time points (e.g. at $T = 10/\mu$), when steady state is reached, as "snapshot".

First, we design experiments to compare unidirectional shared tree scheme (UST scheme for shorthand) vs aggregated multicast scheme with unidirectional shared tree (AM w/UST scheme for short hand). In this set of experiments, each member of a group can be a source and a receiver. Once a multicast session starts up, its core node (or RP) is randomly chosen from the 16 core routers in the network. For aggregated multicast scheme with unidirectional shared tree, the algorithm specified in Section 3.3 is used to match a group to a tree. When members join or leave a group, its aggregated tree will be adjusted according to the matching algorithm. Correspondingly, the routing algorithm A is PIM-SM like routing algorithm which uses unidirectional shared tree.

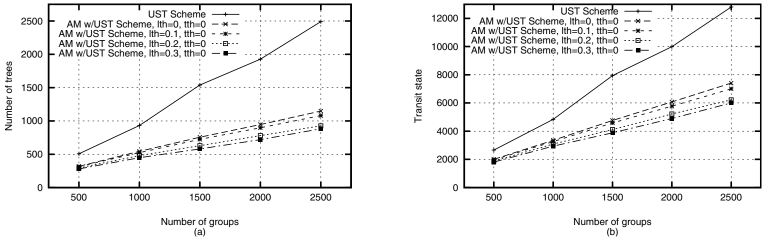


Fig. 3. Results for UST and AM w/UST when only pure-leaky match (tth=0) is allowed

In our first experiment, for aggregated multicast, we only allow pure-leaky match, which means that the tunnelling overhead threshold (represented as **tth**) is 0. We vary the bandwidth overhead threshold (represented as **lth**) from 0 to 0.3. For UST scheme and AM w/UST scheme with different bandwidth threshold, we run simulations to show how the aggregation of aggregated multicast "scales" with the average number of concurrent groups. The results are plotted in Fig. 3. As to the number of trees (see Fig. 3(a)), clearly, for UST scheme, it is almost a linear function of the number of groups. For AM w/UST scheme, as the number of groups becomes bigger, the number of trees also increases, but the increase is much less than UST (even for perfect match ($lth = 0$), the number of trees is only 1150 instead of 2500 for UST when there are 2500 groups). Also this "increase" decreases as there are more groups, which means that as more groups are pumped into the network, more groups can share an aggregated tree. Fig. 3(b) shows us the change of transit state with the number of concurrent groups. It has similar trend to metric number of trees. Transit state is reduced from 12800 to 7400 (above 40% reduction) even for perfect match when 2500 groups come. A general observation is that, when bandwidth overhead threshold is increased, that is, more bandwidth is wasted, number of trees decreases and transit state falls, which means more aggregation. Therefore, there is a trade-off between state and tree management overhead reduction and bandwidth waste.

In our second experiment, for aggregated multicast, we only allow pure-incomplete match, which means that the bandwidth overhead threshold (represented as **lth**) is 0. We vary the tunnelling overhead threshold (represented as **tth**) from 0 to 0.3 and want to look at the effect of tunnelling overhead threshold in the aggregation. Fig. 4 plots the results, which give us curves similar to Fig. 3. However, we can see that tunnelling overhead threshold affects the aggregation significantly: when $tth = 0.3$, and group number is 2500, almost 5 groups share one tree, and transit state is reduced about 70 percentage. When group number increases, we can expect even much more aggregation. The stronger influence of tunnelling overhead threshold on aggregation is not a surprise: the higher the tunnelling overhead threshold is, the more chance for a group to use a small tree for data delivery, the more likely for more groups to share a single aggregated tree.

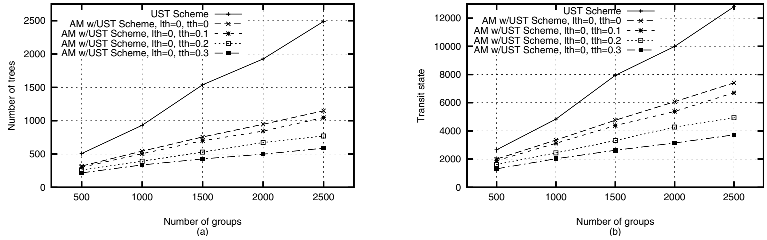


Fig. 4. Results for UST and AM w/UST when only pure-incomplete match (lth=0) is allowed

Our third experiment considers both bandwidth overhead and tunnelling overhead. And the simulation results are shown in Fig. 5. All the results tell what we expect: more aggregation achieved when we sacrifice more (bandwidth and tunnelling) overhead.

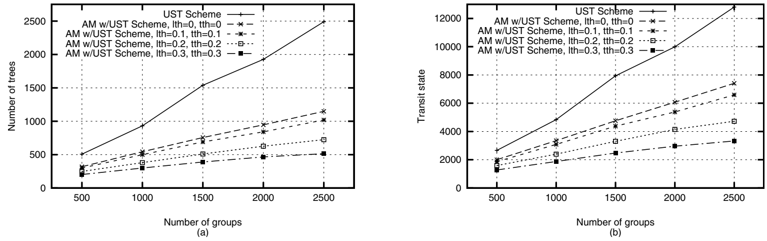


Fig. 5. Results for UST and AM w/UST when both leaky match and incomplete match are allowed

We have shown the results for comparing unidirectional shared tree scheme (UST) vs aggregated multicast scheme with unidirectional shared tree (AM w/UST). Similar results are obtained for source specific tree scheme (SST) vs aggregated multicast scheme with source specific tree (AM w/SST) and bi-directional shared tree scheme (BST) vs aggregated multicast with bi-directional shared tree (AM w/BST). Due to the space limit, we are not going to show the corresponding results for other schemes in this paper. But interested readers can find more results in [4].

From our simulation result and analysis, the benefits of aggregated multicast are mainly in the following two areas: (1) tree management overhead reduction by reducing the number of trees needed to be maintained in the network; (2) state reduction at transit nodes. The price to pay is bandwidth waste and tunnelling cost. The above simulation results confirm our claim while demonstrate the following trends: (1) if we are willing to sacrifice more bandwidth or tunnelling cost (by lifting the bandwidth overhead threshold and tunnelling overhead threshold correspondingly), more or better aggregation is achieved; by "more aggregation" we mean more groups can share an aggregated tree (in average) and correspondingly more state reduction; (2) better aggregation is achievable as the number of concurrent groups increases. The later point is especially important since one basic goal of aggregated multicast is scalability in the number of concurrent groups.

6 Conclusions and Future Work

In this paper, we first gave a classification of multicast schemes, then had a short review of aggregated multicast. For aggregated multicast, we proposed a new group-tree dynamic matching algorithm using tunnelling. We implemented different multicast schemes in SENSE. Through extensive simulations, we compared aggregated multicast with conventional multicast schemes and evaluated its gain over other schemes. Our simulations have shown that significant state and tree management overhead reduction (up to 70% state reduction in our experiments) can be achieved with reasonable bandwidth and tunnelling overhead (0.1 to 0.3), etc.. Thus aggregated multicast is a very promising scheme for transit domain multicast provisioning.

We are now in the process of developing an actual aggregated multicast routing protocol testbed for real application scenarios. The testbed will allow us to better evaluate the state reduction and control overhead.

References

1. vBNS backbone network. *http://www.vbns.net/*.
2. SENSE: Simulation Environment for Network System Evolution. *http://www.cs.ucla.edu/NRL/hpi/resources.html*, 2001.
3. A. Ballardie. Core Based Trees (CBT version 2) multicast routing: protocol specification. *IETF RFC 2189*, September 1997.
4. Jun-Hong Cui, Jinkyu Kim, Dario Maggiorini, Khaled Boussetta, and Mario Gerla. Aggregated Multicast—A Comparative Study. Technical report, UCLA CSD TR No. 020011, February 2002.

5. S. Deering, D. Estrin, D. Farinacci, and V. Jacobson. Protocol Independent Multicast (PIM), Dense Mode Protocol : Specification. *Internet draft*, March 1994.
6. Stephen Deering. Multicast routing in a datagram internetwork. *Ph.D thesis*, December 1991.
7. D. Estrin, D. Farinacci, A. Helmy, D. Thaler, S. Deering, M. Handley, V. Jacobson, C. Liu, P. Sharma, and L. Wei. Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification. *IETF RFC 2362*, June 1998.
8. Aiguo Fei, Jun-Hong Cui, Mario Gerla, and Michalis Faloutsos. Aggregated multicast: an approach to reduce multicast state. *In the proceedings of Sixth Global Internet Symposium(GI2001)*, November 2001.
9. Aiguo Fei, Jun-Hong Cui, Mario Gerla, and Michalis Faloutsos. Aggregated Multicast with Inter-Group Tree Sharing. *In the proceedings of NGC2001*, November 2001.
10. Mark Handley and et al. Bi-directional Protocol Independent Multicast (BIDIR-PIM). *Internet draft: draft-ietf-pim-bidir-03.txt*, June 2001.
11. J. Moy. Multicast routing extensions to OSPF. *RFC 1584*, March 1994.
12. C. Partridge, D. Waitzman, and S. Deering. Distance vector multicast routing protocol. *RFC 1075*, 1988.

New Center Location Algorithms for Shared Multicast Trees

Young-Chul Shim[1] and Shin-Kyu Kang[2]

[1] Hongik University, Department of Computer Engineering, Seoul, Korea
shim@cs.hongik.ac.kr
[2] Aston Linux, Seoul, Korea
cosmos@astonlinux.com

Abstract. Multicast routing algorithms such as PIM, CBT, BGMP use shared multicast routing trees and the location of the multicast tree has great impact on the tree cost and the packet delay. In this paper we propose new center location algorithms and a new center relocation algorithm and analyze their performance through simulation studies. The proposed center location algorithms try to find the geographic center of multicast members considering not only multicast group members but also a few non-member nodes which are carefully chosen. Simulation results show that the proposed algorithms find the better center than existing algorithms in terms of tree cost and packet delay. After many members have joined and/or left the group, the previously chosen center may not be a proper place any more and, therefore, we need to find a new center and build a new tree around this new center. We propose a new center relocation algorithm that determines the moment when the new tree should be built around the new center. The algorithm is based on measured packet delays as well as the parameter indicating how much the group has changed. It not only avoids unnecessary center relocation processes but also prevents the cost and worst packet delay of the tree from significantly deviating from the optimal values. ...

1 Introduction

Multicast is an efficient mechanism for sending packets to a group of receivers and used in many areas[1,2]. To send packets to multicast group members, a multicast routing algorithm builds multicast packet delivery trees among senders and receivers. There are two types of delivery trees: source based trees and shared trees. In the source based tree approach, a shortest path tree is built from a sender to all the receivers and one tree is built for each sender. DVMRP[3] and MOSPF are examples of routing algorithms building source based trees. The disadvantage of this approach is that there are as many trees as the senders and the management of these trees can be very complicated. To solve this problem one shared tree is built among all senders and receivers in the shared tree approach.

This work was supported by ITRC

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 1045–1056, 2002.

CBT(Core Based Trees)[5], PIM-SM[6], and BGMP[7] are routing algorithms in this category. In this approach the location of the center of the shared tree greatly affects the multicast tree cost and the packet transmission delay over the tree and, therefore, the determination of the proper location of the center becomes an important issue. In a dynamic environment where members can join and leave during a multicast session, the center location which may have been optimal in the beginning may not be so anymore after many membership changes. So in case of the dynamic environment, the relocation of the center also becomes an important issue.

The center location algorithms can be divided into three categories depending upon what network nodes are considered as candidates for the center. In the first category, all the network nodes can become candidates for the center and the best node is chosen as the center. With this method, the optimal center location can be found but because too many packets are exchanged among all the network nodes, it is never a practical solution. In the second category, only the multicast members are considered as candidates. This approach incurs the least overhead but because only the members are considered, the chosen center location can be far from being optimal. The last category stands between the first and the second. In this category, not only the members but also some carefully chosen non-member nodes are considered for the center and the best node among them is chosen as the center. The method of choosing non-member nodes that will be considered as candidates affects the overhead of the center location algorithm and the quality of the chosen center.

We propose a new center location algorithm called GeoCenter(Geographic Center). The idea is that we try to find the geographic center of the multicast members in the Internet map and this geographic center becomes the center of the multicast tree. This geographic center can become the member or non-member router. We introduce three algorithms GeoCenter1, GeoCenter2, and GeoCenter3 depending upon the method of finding the geographic center. The proposed algorithms try to minimize the packet delay and the tree cost. Then we consider a dynamic case and propose a new algorithm for relocating the center as the membership changes. The center relocation process is such a costly one that its execution should be limited only to unavoidable cases. The new algorithm determines the moment when the new tree should be built around the new center. This algorithm is based on measured packet delays as well as the parameter indicating how much the group has changed. It not only avoids unnecessary center relocation processes but also prevents the cost and worst packet delay of tree from deviating too much from the optimal value. We analyze the performance of the center location and relocation algorithms through simulation.

The rest of the paper is organized as follows. Section 2 surveys related work. Sections 3 and 4 describe our algorithms for center location and relocation, respectively. Section 5 presents simulation results and is followed by the conclusion in Section 6.

2 Related Work

In this section we first present algorithms that have been proposed for the center location. Before introducing these algorithms we give the definition of the tree cost and explain weight functions that have been used in those algorithms. The tree cost is the sum of the cost of each link in the tree. The link cost can be the actual monetary value of that link, bandwidth, delay, etc. But in this paper we set the link cost to be 1 for every link. A weight function is calculated for each center candidate and the resulting values are compared to select the best one. We introduce the definitions of some weight functions in the following[8]. In the definitions, S is the set of all the senders and members, u and v represent either a sender or a member, root is the candidate for the center, d(u,v) is the distance between u and v and deg(u) is the degree of u.

$$Actual\ Cost = number\ of\ links\ in\ tree\ rooted\ at\ root\ .$$

$$Max\ Dist = \max_{u \in S}\ d(root, u)\ .$$

$$Avg\ Dist = \frac{1}{|S|} \sum_{u \in S} d(root, u)\ .$$

$$Max\ Diam = \max_{u \in S}\ d(root, u) + \max_{v \in S, v \neq u}\ d(root, v)\ .$$

$$Est\ Cost = \frac{Est\ Cost_{min} + Est\ Cost_{max}}{2}\ .$$

$$\begin{aligned} where \quad Est\ Cost_{min} &= \max_{u \in S} d(root, u) + \\ &\quad number\ of\ duplicate\ distance\ nodes\ in\ S \\ Est\ Cost_{max} &= \begin{cases} \sum\limits_{u \in S} d(root, u) & if\ |S| \leq deg(root) \\ [\sum\limits_{u \in S} d(root, u)] - [|S| - deg(root)] & otherwise \end{cases} \end{aligned}$$

Now we introduce several center location algorithms. The OCBT(Optimal Center-Based Tree) algorithm calculates the actual cost of the tree rooted at each node in the network and selects the one which gives the lowest maximum delay over all the roots with the lowest cost. The MCT(Maximum-Centered Tree) algorithm selects the node with the smallest $MaxDist$ value. The ACT(Average-Centered Tree) algorithm chooses the node with the smallest $Avg\ Dist$ value. The DCT(Diameter-Centered Tree) algorithm selects the node with the lowest $Max\ Diam$ value. These four algorithms belong to the first category of center location algorithms.

The RSST(Random Source-Specific Tree) algorithm chooses the center randomly among the senders and is used in CBT and PIM. In the MIN-MEM(Minimal Member Tree) algorithm, each member or sender node calculates the weight function of the multicast tree rooted at itself and exchanges the calculated value with other nodes. The node with the lowest weight function value becomes the center. The weight function can be any of the five functions explained above. These two algorithms belong to the second category.

In the HILLCLIMB algorithm, a randomly chosen temporary center calculates the weight function of the tree rooted at itself and all the routers directly connected to the temporary center do the same calculation. If the temporary center has the lowest value, it becomes the center. Otherwise the node with the lowest value becomes the temporary center and compares the weight function value with its direct neighbors. This process is continued until the node is found such that its value is lower than those of its direct neighbors or the distance from the original temporary center to the current temporary center reaches a certain threshold. This algorithm belongs to the third category. The problem of this algorithm is that it just finds the locally optimal point among nodes within a limited distance from the original center.

In a dynamic environment, a center can be relocated by applying any of the above algorithms after some membership changes have occurred. The biggest issue here is when to apply the center location algorithm again. Thaler and Ravishankar introduce the parameter Δ defined as follows[9]:

$$\Delta = 1 - \frac{|G_0 \cap G_i|}{max(|G_0|, |G_i|)} .$$

where G_0 is the original group membership, G_i is the current group membership, and Δ indicates the amount by which the group has changed. They propose to recalculate the center location when Δ reaches 90%. They show that when 90% of the membership has changed, the tree cost has likewise degraded about 90% of the way toward a randomly centered tree. But we show that their algorithm does not improve the quality of the tree center at all in some cases and, therefore, incurs unnecessary center relocation processes.

3 New Algorithms for Center Location

In this section we introduce 3 new center location algorithms: GeoCenter1, GeoCenter2, and GeoCenter3. These algorithms pick the center based upon the information on routes between members. The route information from a node A to a node B is the list of all the routers visited on the path from A to B and this information can be obtained by using the program called traceroute or the IP route record option. In the GeoCenter1 algorithm, each member finds the route information to all the other member nodes, compiles all the routers appearing in the routes, and sends this information to a temporary center. Upon receiving the route information from all the members, the temporary center finds the routers that appear most frequently in the route information. If there are several such routers, one router is selected randomly. This selected router and the member nodes become the candidates for the center. The center is chosen as the node which has the lowest $Max\ Dist$ value among these candidates.

GeoCenter2 and GeoCenter3 take different approaches in selecting non-member candidates. In the GeoCenter2 algorithm, each member first collects route information from all the other router as in GeoCenter1 but, when compiling this information, records not only the addresses of each router in the route information but also the number of times a router appears in the route information. This information is sent to the temporary center. In the GeoCenter3 algorithm, each member finds the midpoints on the path to other members and sends the list of these midpoints to the temporary center. The way the temporary center selects the center is the same as in GeoCenter1.

Based upon the explanation given in the above, we now present each algorithm in detail.

GeoCenter1

① When a multicast group is created, a temporary center is chosen arbitrarily.
② The temporary center sends a probe message to each member. The probe message also contains the list of the multicast group members.
③ Upon receiving the probe message, each member finds the route information to all the other members using either the traceroute program or the record route IP option. At the same time the member measures the packet delay to other members and records the largest packet delay as its weight function value.
④ Each member compiles the list of nodes appearing on the path to other members from itself. It sends this list and its weight function value to the center.
⑤ Upon receiving the list from all the members, the temporary center selects a node(s) that appears most frequently in the lists. If there are many such nodes, one or more nodes are randomly picked. If the picked node(s) are a member, go to step ⑧.
⑥ The temporary center sends a probe message to the selected non-member candidate(s).
⑦ The non-member candidate measures the packet delay to all the members, records the largest delay as its weight function value, and sends this value to the temporary center.
⑧ The temporary center selects the node which has the lowest weight function value as the center.

GeoCenter2

GeoCenter2 is the same as GeoCenter1 except the step ④. The member nodes send not only the address of nodes on the path to the other member nodes but also the visit counts of such nodes.

GeoCenter3

①②③ The same as in GeoCenter1.
④ Each member finds the midpoints on the path to other members and sends the list of midpoints and its weight function value to the temporary center.
⑤ The temporary center adds up all the visit counts for each node appearing in the lists received from the member nodes. It picks the node(s) with the largest accumulated visit counts. If the picked node(s) is a member, go to step ⑧.
⑥⑦⑧ The same as in GeoCenter1.

4 The Center Relocation Algorithm in a Dynamic Multicast Environment

In this section we explain the algorithm for relocating the center after membership changes have occurred many times. After many members have joined and/or left the group, the center that was carefully chosen in the previous time may not be a good place any more. The quality of the tree may have deteriorated during the membership changes and, therefore, the tree cost and the maximum delay may have become too high compared with the optimal tree. As already explained in the previous section, the most important issue in the center relocation is the determination of the moment when the center location algorithm is applied. Because the center relocation process requires not only the determination of a new center location but also building a new multicast delivery tree around this new center, it is a very costly process. In reality, building a new tree will consume more time than calculating a new center location. So it is imperative to minimize the numbers that the multicast delivery tree is rebuilt.

As we described in Section 2, Thaler and Ravishankar introduce a parameter Δ indicating the amount by which the group has changed and used this parameter to determine when to calculate the new center location. They propose to calculate the new center location when Δ reaches 90%. They also show that the time interval which it takes for Δ to reach from 0 to 90% roughly corresponds to two to three times of the average connection duration of a member in a multicast group. But as we will show with simulation, if the area in which members are located does not change very much and members are uniformly distributed in this fixed area, the center relocation using only Δ does not improve the tree cost and the packet delay and, is unnecessary in many cases.

Another parameter we can use to determine when to calculate the new center location is the maximum delay from the center to the member nodes. When the center location is calculated, the maximum delay at that moment is recorded as Prev_Max_Delay. When a new member joins the group, the delay from the current center to this node is calculated and compared with Prev_Max_Delay. If the delay to this new node exceeds a certain constant times of Prev_Max_Delay, a new center location is calculated. Using this method, the unnecessary recalculation of the center in the case of just using the parameter Δ can be avoided because the center will be recalculated only if the proof that the quality of the tree has deteriorated enough is obtained. This method of measuring the delay to a joining member and comparing against the Prev_Max_Delay works if the area where members are located remains the same, moves, or gets larger. But this method does not work if the member distribution area gets smaller. If the size of the area gets smaller, the delay to a new member will rarely exceed the previously measured Prev_Max_Delay value. But the quality of the tree may have been deteriorated compared with the optimal tree.

We propose a new algorithm for determining when to recalculate the center location and when to actually rebuild the tree around the new center. The proposed algorithm uses both the parameter Δ and the delay to a new member. We assume that the multicast routing algorithm enables the center to be notified of all the join events of new members and calculate the delay to these new members. Now we explain our algorithm in detail as follow.

① The location of a center is calculated.
② The multicast delivery tree is built around the calculated center. The Prev_Max_Delay and Curr_Max_Delay values are initialized to be the value of the maximum delay of this new tree. Set G_0 and G_i to be the current set of members.
③ Wait until a membership changes. If the membership change is a join event, calculate the delay from the current center to this new member. Update the Curr_Max_Delay to be the maximum of the delay to this new node and the current value for Curr_Max_Delay. If Curr_Max_Delay is greater than C1 * Prev_Max_Delay, go to step ①.
④ Update G_i and the value of Δ. If this value does not exceed C2, go to step ③. Otherwise calculate the location of a new center. Assuming this new center, calculate the maximum delay of the new tree and set this value to be Opt_Max_Delay. If Curr_Max_Delay is greater than C1 * Opt_Max_Delay, go to step ②. Otherwise, set G_0 and G_i to be the set of current members and go to step ③.

In the above algorithm C1 determines the extent to which the worst case packet delay of the current tree is permitted to exceed the worst case packet delay which was calculated when the center was determined in the most recent time. If C1 is 1.4, the excess up to 40% is permitted. If Δ reaches C2, it means that C2 * 100% of members have changed. In the algorithm given in [9], C2 was set to be 0.9. The algorithm calculates a new center location and builds a new tree around this new center if the delay to a new joining member is bad enough compared with the maximum delay which was calculated when the tree was built in the last time. But the area where the members are distributed gets smaller, this condition will be rarely satisfied. So after we have seen enough changes in the membership, we calculate the new center location and also calculate the new tree around this new center. If we see that the quality of the current tree is bad enough compared with this new tree, we actually rebuild the tree around the calculated new center .

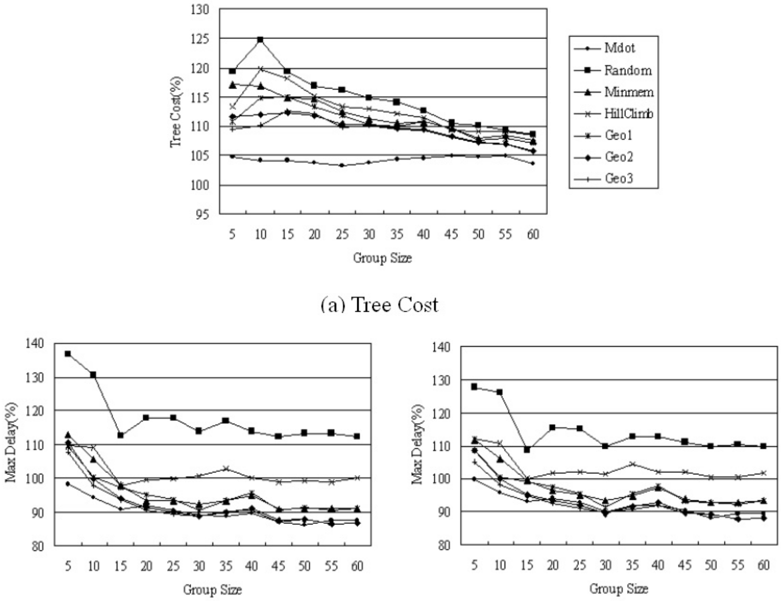
5 Experimentation Results

In this section we present and analyze simulation results for our center location and relocation algorithms. We first present simulation results for our center location algorithms assuming that the set of members and senders are fixed and compare their performance with other center location algorithms. Then we show simulation results for our center relocation algorithm in the environment where members join and leave.

For the experimentation we used NS(Network Simulator) developed in UC Berkeley and ran the simulation on Linux 5.1 platforms. Algorithms were implemented with TCL and network topologies were generated with the GT-ITM(Georgia Tech Internetwork Topology Models) provided in the NS.

5.1 Experimentation Results for Center Location Algorithms

In this subsection we compare the proposed center location algorithms with other algorithms through simulation changing the multicast group size, the network size, and the average number of links for a node in the network. We measure the tree cost and packet transmission delay of the multicast trees built using various center location algorithms. We compare the proposed algorithms with OCBT, MDOT(Minimum Delay Optimal Tree), Random, MIN-MEM, and HILLCLIMB algorithms. The OCBT algorithm is optimal in terms of the tree cost. The MDOT considers all the nodes as the candidates for the center and picks the node such that the tree built around this node has the lowest $MaxDist$ weight function value. If several nodes have the same lowest $Max\ Dist$ value, the node with the lowest tree cost is selected. This algorithm is optimal in terms of tree costs when the shared trees are unidirectional such as in PIM-SM but may not be optimal in case of bi-directional shared trees such as in CBT. The Random algorithm chooses the center randomly. These three algorithms, OCBT, MDOT, and Random, are not practical but considered here just for the comparison. MIN-MEM and HILLCLIMB algorithms are practical solutions and shown to find a good center[8,9]. For the experimentation we assume that all the senders are also members and for each simulation we perform 100 experiments and take the average as the result.



(a) Tree Cost

(b) Delay of a unidirectional tree

(c) Delay of a bidirectional tree

Fig. 1. Effects of group size on algorithms

A group size is the number of member nodes in a multicast group. In the first experimentation we assumed that there were 100 nodes in the network and the average number of links for nodes was 4. We changed the group size from 5 to 60 and measured the tree costs and the packet delays. Figure 1 (a) shows the tree costs of various algorithms and the tree cost is represented as the ratio to OCBT. The methods for measuring the packet delay become different depending on the type of shared trees: unidirectional trees or bidirectional trees. In unidirectional shared trees, packets are sent to the center and then distributed to all the members. But in bidirectional shared trees, packets are sent along the shortest path on the shared tree from the sender to members and in many cases may not pass the center. Figures 1 (b) and (c) show the packet delays of various algorithms and the packet delays are represented as the ratio to OCBT. In the figures some algorithms sometimes show better packet delay better than MDOT and this can be possible because MDOT is optimal when the packet delays are measured between the center and the receivers not between senders and receivers. The figures show that the tree cost of GeoCenter2 and GeoCenter3 stays within 112% of OCBT and is better than MIN-MEM and HILLCLIMB and the packet delay of GeoCenter2 and GeoCenter3 is comparable to MDOT and always lower than other algorithms. We see that GeoCenter2 and GeoCenter3 algorithms show better result than GeoCenter1 because former algorithms find better geographic centers than GeoCenter1.

Next we summarize the results for the second and third sets of simulations without showing the figures because they were similar to Figure 1. The second set of experiments was performed varying the network size that is the number of nodes in the simulated network. The average number of links for nodes was set to 4 and the group size was assumed to be 20% of the network size. The tree cost of GeoCenter2 and GeoCenter3 was 12% higher than OCBT in the worst case but always better than MIN-MEMB and HILLCLIMB. The packet delay of GeoCenter2 and GeoCenter3 was comparable with MDOT, even better than MDOT at some points, and constantly better than other algorithms.

The third set of experiments was performed varying the average number of links of a node in the network. The network size and the group size were assumed to 100 and 20, respectively. The simulation showed that the tree cost GeoCenter2 and GeoCenter3 stayed within 114% of OCBT and always better than MIN-MEMB and HILLCLIMB. The packet delay of GeoCenter2 and GeoCenter3 was very similar to MDOT and always better than other algorithms.

From the above three sets of experiments, we conclude that two of the proposed algorithms, GeoCenter2 and GeoCenter3, achieve near optimal packet delay compared with the MDOT algorithm while not incurring too much increase on the tree cost compared with the OCBT algorithm.

5.2 Experimentation Results for the Center Relocation Algorithm

In this subsection we consider a dynamic environment where members can join and leave during the lifetime of a multicast session.

Thaler and Ravishankar introduced the parameter Δ and proposed to calculate the new center and rebuild the tree as Δ reaches at some fixed value[9]. We first show that if the area in which the members are located is fixed and the members are uniformly distributed in this area, their simple method of just using the Δ value does not improve the quality of trees at all. We ran experiments with a network of 200 nodes. The members were uniformly distributed in this network and their average number was 40. We compared the quality of trees of two cases. In the first case the center location is never recalculated and in the second case the center location is recalculated when Δ reaches 90%. The simulation results showed that recentering and rebuilding a tree with just using Δ did not improve the quality of trees at all and in some cases gave worse performance. So we conclude that if members are uniformly distributed in a fixed area, we need not recalculate the center location. This is because the center that was calculated in the beginning remains to be near optimal if the distribution of the members remains uniform even though members join or leave the multicast group. From the experiment we can see that the packet delay remains to be within 140% of the optimal value, so if we set C1 to be 1.4 in our center relocation algorithm, the center need not be moved and, therefore, unnecessary overhead can be avoided. But the algorithm by Thaler and Ravishankar using only Δ regularly changes the center location but does not improve the quality of the multicast tree.

Now we consider three cases where the area in which members are distributed changes and show how our center relocation algorithm explained in the previous section performs. In the first case the area expands, in the second case the size of the area remains the same but the area moves gradually, and in the last case the size of the area becomes reduced.

Figure 2 shows the simulation results when the area expands. Each figure has two graphs. The first graph shows the result when the center is calculated once in the beginning and is never recalculated. The second graph describes the result when the center is recalculated by the proposed algorithm using both Δ and the worst case packet delay measurement. In the experiments GeoCenter3 algorithms was used to determine the center location. The points on graphs are represented

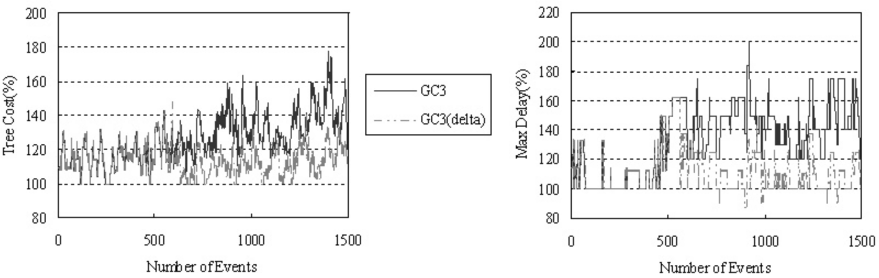


Fig. 2. Center relocation in an expanding area

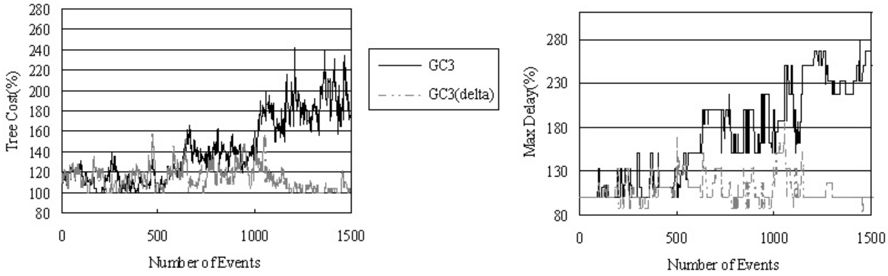


Fig. 3. Center relocation in a moving area

as the ratio to the value of optimal tree generated using the OCBT algorithm at each measurement point. The figure shows that the tree cost gradually increases without the center relocation algorithm as the area expands with the center recalculation. But if we use the proposed center relocation algorithm, the tree cost never becomes 30% higher than the optimal value calculated with the OCBT algorithm at each point. The figure also shows that the packet delay becomes almost 1.8-2 times of the OCBT tree without the center recalculation but rarely becomes 20% higher than the OCBT tree if we use the proposed center relocation algorithm.

Figure 3 shows the simulation results when the area is moving and Figure 4 shows the simulation results when the area gets reduced. They show the same result as in the case when the area gets expanded.

From these simulation results we see that the algorithm using just the Δ parameter regularly recalculates the center location and actually rebuilds the multicast tree without making much improvement on the tree quality in the case that the area in which members are distributed is fixed. We note that rebuilding a tree is a very costly process. But the proposed algorithm uses both the Δ parameter and the measurement data of the worst case packet delay and can avoid unnecessary rebuilding of the multicast tree in this case. In the cases where

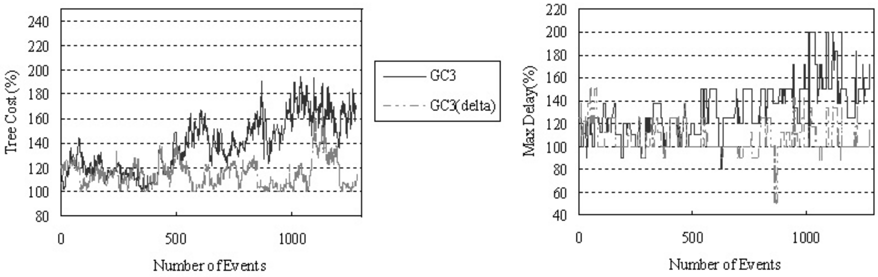


Fig. 4. Center relocation in an reducing area

the area gets expanded, moves, or becomes reduced, the proposed algorithm generates multicast trees of reasonable quality. And by properly adjusting the value of C1, which is the multiplier on the previously measured maximum packet delay and determines when the center should be recalculated, we can bound the maximum packet delay within a certain limit of the maximum delay of the OCBT.

6 Conclusion

In this paper we proposed new center location algorithms and a new center relocation algorithm for multicast routing alogrithms building shared trees and analyzed their performance through simulation.

The proposed center location algorithms try to find the geographic center of multicast members considering not only multicast group members but also a few non-member nodes that are carefully chosen. We built multicast trees around the centers found by our algorithms and we found that these trees had slightly higher tree cost than the cost-optimal tree, the similar packet delay as the delay-optimal tree, and constantly better cost and packet delay than the trees built around the centers found by algorithms proposed by other researchers.

Then we considered a dynamic environment where members could join and leave a multicast session and proposed a center relocation algorithm which determined the moment when the new tree should be built around the new center. The algorithm is based on measure packet delays as well as the parameter indicating how much the group membership has changed. Our algorithm not only avoids unnecessary center relocation processes but also prevents the cost and worst packet delay of the tree from deviating too much from the optimal values.

References

1. T.A. Maufer: Deploying IP Multicast in the Enterprise. Prentice Hall. (1997)
2. B. Quinn: IP Multicast Applications: Challenges and Solutions. Internet Draft drft-ietf-mboned-mcast-apps-01.txt. (June 1999)
3. D. Waitzman, C. Partridge, and S. Deering: Distance Vector Multicast Routing Protocol. RFC 1075. (1988)
4. J. Moy: MOSPF: Analysis and Experience. RFC 1585. (1994)
5. B. Cain, Z. Zhang, and A. Ballardi: Core Based Trees Multicast Routing: Protocol Specifcation. (1998)
6. D. Estrin et al: Protocol Independent Multicast-Sparse Mode: Protocol Specifiction. RFC 2362. (1998)
7. S. Kumar et al: The MASC/BGMP Architecture for Inter-Domain Multicast Routing. ACM SIGCOMM Conference. (August 1998)
8. D. Thaler and C. Ravishankar: Distributed Center-Location Algorithms: Proposals and Comparisons. IEEE Infocom. (1996)
9. D. Thaler and C. Ravishankar: Distributed Center-Location Algorithms. IEEE Journal on Selected Area in Communications, vol. 15, no. 3. (1997)

A Multicast FCFS Output Queued Switch without Speedup

Maurizio A. Bonuccelli and Alessandro Urpi

Dipartimento di Informatica, Università di Pisa,
Corso Italia 40, 56100 Pisa, Italy. {bonucce,urpi}@di.unipi.it

Abstract. In this paper we propose an architecture for an output queued switch based on the mesh of trees topology. After establishing the equivalence of our proposal with the output queued model, we analyze its features, showing that it merges positive features of the input queued switches (specially their implementability) with all the characteristics typical of output queued ones. Moreover, such an architecture is able to easily and efficiently manage multicast traffic, which is becoming extremely important in networks with traditional communication services integrated in.

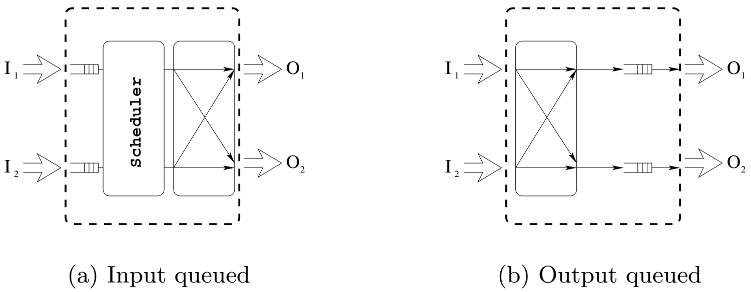
1 Introduction

Internet is evolving to an integrated services network with a large number of users that exchange huge amounts of data, making the efficiency of the switching phase increasingly critical ([1,2,3,4]). This is even more evident since large parts of Internet are circuit switched (SONET[1], just to cite one name), and since link speed is rapidly increasing (for example, 40 Gb/s at OC768c or 160 Gb/s at OC3072), making routers/switches a serious bottleneck. At a suitable level of abstraction, a switch is a box connecting n source inputs that want to exchange messages with m destination outputs. The system is synchronous, and the time is slotted. Without loss of generality, we can think of messages as fixed size cells that arrive at the system at the beginning of each slot, and are processed during the time interval. Since we assume that message destinations are independently chosen by each input without rules, it can happen that more inputs want to communicate with the same output at the same time, causing a potential collision. Such an event should be avoided, because it results in the loss of all the cells involved in it, and in the retransmission of all of them from the originating source. Competing cells need to be stored in a memory, and to be serialized in some way, in order to keep busy outputs with queued cells for and to avoid collisions.

There has been a deep investigation in buffered switches during the last years, that leaded to fundamental results. One of the first proposed solutions ([5,6]) was to put a shared memory between inputs and outputs where to store incoming cells and to forward a suitably chosen subset of them. While such an architecture

[1] http://www.sonet.com

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 1057–1068, 2002.



(a) Input queued

(b) Output queued

Fig. 1. Different switch architectures.

is quite simple and practical for systems operating at less than 20 Gb/s, it has many problems, the most penalizing is perhaps the memory access time ($n + m$ accesses should be granted at every cycle).

A natural step to move then was to introduce a queuing system, that led to input queued (Fig. 1(a)) and output queued (Fig. 1(b)) switches. The former is the implementation of the very simple idea that every cell, at the arrival to the switch, should be immediately buffered (with a queue for every input), and then a scheduler will choose in every cycle a set of non conflicting cells (namely, cells bound for different outputs) to forward through a nonblocking interconnection network, for example a crossbar. Easy to implement, the architecture was shown to suffer of limited throughput if a FIFO strategy is used in the queues: conflicting cells in the head of the queues may block other cells that would be free to pass through the switch, causing a performance loss known as *head of line* (HOL) blocking, limiting the throughput of the system to $\sim$ 58.6% assuming i.i.d. arrivals ([7]).

Moving the queues at the output ports results in efficient switches that don't block cells if their destination is idling, able for this reason to provide quality of service ([1,8]). This is not a solution, since such an architecture is clearly equivalent to the shared memory one, and its problem is again scalability: each queue must be able to serve up to n requests per time slot. This introduces the need for a speedup of the switch of a factor $n + 1$, limiting its implementability to scenarios with few input ports and quite slow links. In order to achieve scalability without performance problems, virtual output queued switches were proposed ([9,2]). Such an architecture avoids HOL blocking by having in each input a different queue for each output. It is clear that the scheduling phase is now critical: a set of cells must be selected for transmission at every time slot to maximize performances. It was shown ([10,11,12,13]) that there exist scheduling algorithms able to exploit a throughput of the 100% and also to avoid starvation of cells ([14]). However, such algorithms have several drawbacks, that can be so classified:

- **Complexity:** an optimal scheduling can be found solving a matching problem on bipartite graphs ([15]), or finding a decomposition of stochastic matrices (see [16] for the switching case). The weak point of these approaches is their complexity; the best known matching algorithm runs in $O(N^2 \log_2(N))$ time in the worst case ([17]), while the second method has been proved useful to implement, at most, 4×4 switches ([18]).
- **Throughput:** with approximate algorithms it is possible to overcome the complexity problem ([2,15,19,13]). Behind this approach there are good simulation results ([15,20]), and a proof that, if traffic reaches a steady state (i.e. there is always a cell that must be sent to every output), the behavior of these algorithms is optimal ([13]). But with *bursty traffic* ([21]) such a stability is never reached ([20]).
- **Performance guarantees:** despite some results on the bounds in queues average sizes and on average delays in input queued switches have been recently found ([22]), it is not yet clear how to offer quality of service in such a class of switches. This justifies the research on output queued like architectures, in order to obtain guarantees on the offered service.

Combined input-output queued switches are another interesting architecture proposed as a trade-off between input and output queuing: there are queues both in the inputs and in the outputs, and a speedup of k is used, in the sense that it is possible to transfer k cells from every queue in the inputs to the desired queue in the outputs at every time slot. In [23,24] it was proved that a speedup of 2 is enough and necessary to emulate an output queued switch with a queuing policy that varies in a well known class. Unfortunately a locally optimal scheduling does not guarantee network optimization: in [25] it is shown that input queued switches with high performance scheduling algorithms, efficient in isolation, cause unbounded delay of cells when put in a network. In order to avoid this problem, and to offer quality of service (QoS), it is then important to have practical solutions resembling output queued switches. Parallel architectures are an encouraging alternative ([26,27,28,29,30]).

In this work we take a completely different approach, and propose an output queued switch obtained by parallelizing the “classical” architecture, having very interesting features like high compositional power (i.e. it is easy to create a greater switch by using smaller ones), no speedup required (in a sense that will be cleared later), real implementability and efficient multicast management.

The paper is organized as follows. Section 2 introduces the notation we will use throughout the paper. Section 3 outlines the idea at the very base of our proposal, relating it with a well known architecture. In Sect. 4 the topology of the mesh of trees is presented, together with some of its most important features. In Sect. 5 we present a new architecture for a switch, proving that it is equivalent to an output queued one. Finally, we conclude in Sect. 6 summarizing our work and proposing future directions.

2 Definitions

The following concepts and terms are very important through the paper:

Number of ports: without loss of generality, the switches are supposed to have n inputs and n outputs,

Names: I_i is the i^{th} input, O_j is the j^{th} output; Q_i is the queue at the i^{th} input (output) in an input (output) queued switch, and L_i is its size,

Acronyms: *IQ* means Input Queued, *OQ* is Output Queued, while *VOQ* is Virtual Output Queued and *CIOQ* is Combined Input-Output Queued,

Mimicking: as defined in [24], a switch S mimics another switch $S^{'}$ if, for any arrival pattern and independently of the switch size, the outputs are exactly the same. In [30] the definition is extended by considering a possible queuing delay for the cells, i.e. the outputs of the two switches are the same but with a temporal shift caused by queuing. So, an architecture X mimics an architecture Y with a delay of $f(n)$ if, under the same arrival process, the outputs of X at time $t + f(n)$ are the same of Y at time t.

3 A First (Impractical) Step

We begin by presenting a new point of view of an *OQ* switch. Later in this section the same intuition will be presented from a different perspective.

Let us assume we have a $n \times n^2$ crossbar[2] and that each cell is associated with an integer representing its arrival time (a time stamp). Then, we can think of splitting the queues in the output ports in n different queues, one for each input, like in Fig. 2. Such an architecture can be thought of as the complement of a *VOQ* switch, and it should not be hard noting that it can perfectly emulate an *OQ* switch. In fact, assuming a FIFO strategy, the division in n queues is equivalent to the distribution of the cells in queues, sorted by sender. Q_i in an *OQ* switch with a speedup of n, would contain all the cells sent to output i, sorted by the time of arrival to the system, with simultaneous arrivals serialized with a specific rule (for example smaller index of sender first, or randomly). In this way, in the n queues at the i^{th} output, there is a double sorting: by arrival time and by sender. Then, if the S_i element chooses the oldest cell from all the queues, breaking ties with the same rule that would have been used in the target *OQ* switch, we perfectly emulate it.

The proposed architecture apparently does not require any speedup to achieve an *OQ* switch behavior emulation. Actually there is a logarithmic factor to be accounted for. In fact, assuming to be able to compare n time stamps in only one cycle is unrealistic (specially for very large n, that is our final target). The best thing we can do is to use a comparing tree, with $\log_2(n)$ stages of parallel comparisons. Such a logarithmic factor must be paid off in terms of scheduling iterations, in the worst case, also in almost every proposed *VOQ* switch ([2] for

[2] Actually, a full crossbar is not necessary. A structure containing n selectors or a sorting network would be enough, but it is easier to imagine a crossbar

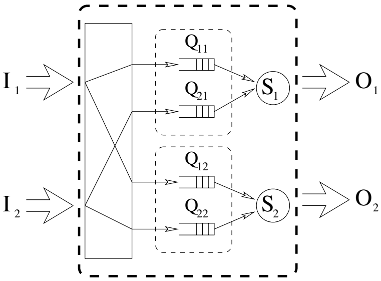


Fig. 2. Output queued switch?

PIM, [15] for iSLIP, [31] for iLPF, just to cite some very popular proposals). In *IQ* and *VOQ* switches, it is usually assumed that these computational steps can be done during a time slot (thus limiting the power of scheduling algorithms). In our proposal, we can avoid this delay with a very simple pipelining technique. In fact, it is not necessary to wait for the entire comparison to be over before starting a new one but, because of the tree structure of the comparison part, each element (leaf or internal node) can compare two cells, and forward the oldest to its parent (the root must send the oldest outside the switch). Thus, as soon as an element finishes its work, it can start again for another round, waiting only if its successor in the tree (its parent) is not ready to receive. Thus it is possible to perform $\log_2(t)$ comparisons (one for level) in parallel. The latency of the cell in the switch is proportional to the logarithm of the switch size but the throughput of the system will not suffer from this.

The introduction of a pipelined part in a switch is not a new concept: in [32] the scheduler for a *VOQ* switch is improved with this technique, but the whole system is very different (and more complicated) from the one presented here. We make another step in our description, in order to have a system easier to implement. It is possible to come to the same idea also by starting from well known results. In [33], the architecture of an *OQ* switch called *knockout* was presented. Each input is connected to one bus, and each output is connected to every bus. We can think of the output modules as single queues, and still have the mentioned speedup problem. We can also think of increasing the number of queues, in order to avoid speedups by increasing the cell loss probability (namely, the probability of dropping conflicting cells). Of course, by putting one queue for every input in each output module , there is no cell loss (at this stage), and the architecture is very similar to the one we sketched. It is also possible to put less queues (say L), preceded by a statistical multiplexer that just chooses L cells, if there are more, and discarding the others. A buffering scheme that uses several (L) FIFO queues as just one queue with L inputs and one output in a knockout switch makes it acting like an *OQ* switch without requiring any speedup. It was shown that $L = 8$ queues are sufficient to reduce the loss probability to 10^{-6} for an arbitrarily large switch size n ([33]). However we are interested in avoiding cell loss (and then in using n queues without multiplexer). The architecture, conceptually interesting, has many problems, like number of busses too high when the number of inputs grows and a number of crossing points not feasible when there are many outputs. Moreover, there is a spatial speedup to pay: implementing N adjacent memories is not so different from implementing one with a (temporal) speedup of T. In Sect. 5 we will see why the architecture proposed in this paper can be more practical, while potentially having the same problems.

4 Mesh of Trees

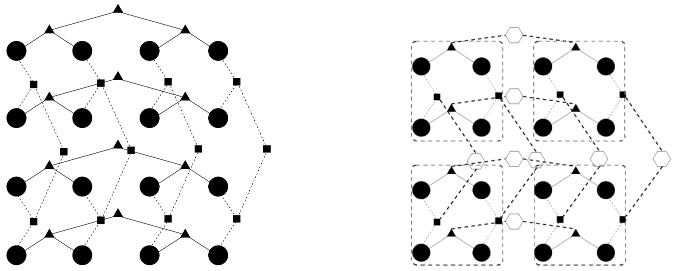
We present here a well known topology called mesh of trees, recalling only what is helpful to our aim. For more details the reader can refer to [34].
An $N \times N$ two-dimensional mesh of trees is a structure obtained from a $N \times N$ mesh (or two-dimensional array) by adding nodes in order to form a complete binary tree for every row and every column, with the nodes of the mesh as shared leaves (see Fig. 3). There is also an interesting recursive definition of the topology: given four $\frac{N}{2} \times \frac{N}{2}$ meshes of trees it is possible to combine them in a $N \times N$ one just by using the four smaller meshes as elements of a 2×2 mesh, and combining the $4N$ roots pairwise adding $2N$ new roots (for a practical example see Fig. 3(b), where the nodes to be added are represented by hexagons).
The total number of nodes in a $N \times N$ mesh of trees is $4N^2 - 2N$. Communications between root nodes of column trees and root nodes of row trees are interesting in several ways. First of all they have a fixed length of $2\log_2(N)$ hops. Moreover, if we label each destination node with the binary representation of a number between 0 and $N-1$ (of course a different label for each different node), the routing of a message through the mesh of trees is very simple (i.e. the topology has a *self-routing* property). For example, nodes at i^{th} level will forward the message to their right son if the i^{th} digit of the label is 0, to their left son otherwise. The leaves work as interchange points, and they just have to forward the message from the column tree to the row tree they belong to. The communication is then logically divided in two steps:

1. a *selection* phase, in which the message is directed to the right row,
2. a *gathering* phase, in which the message is conveyed to the desired root.

In terms of hardware complexity it is clear that each node, if only communication is needed, is very simple. Implementability of meshes of trees in single chips was widely studied (e.g. see [35]). In the next section, we will see how to combine meshes of trees and the switch architecture we proposed in Sect. 3.

5 The New Architecture

It is possible to produce a $n \times n$ OQ switch equivalent to the one we presented in Sect. 3, by means of a $n \times n$ mesh of trees. Assume to associate each input to a column tree root, and each output to a row tree root. In this way, a cell from



(a) 4 × 4 mesh of trees

(b) Composition of smaller meshes

Fig. 3. Two views of a 4 × 4 mesh of trees

input i to output j can be seen as a communication between roots, exactly like those presented in Sect. 4. Thus, the selection stage is exactly equivalent to the crossbar-like element in Fig. 2, while the gathering stage, choosing the oldest cell in case of contention, is just a comparing tree. We can think to put the queues in the leaves: in this way they would become very simple elements encapsulating a queue. It is easy to see that the two architectures are equivalent. Later in this section, a formal proof of the above mentioned mimicking will be presented.
The logarithmic factor in the selection stage can be amortized in the same way we did in the comparing phase: up to $\log_2(t)$ communications can be present in parallel in the column trees, for a total of $2\log_2(t)$ communications at most that can be done in parallel. In the remainder of this section, we shall assume infinite size queues, and we use the following additional notation:

Memories: in the mesh of trees, for each output, the memory is divided into *queuing memory* (in the leaves) and *tree memory* (up to one cell can be stored in each internal node in the row tree while waiting to exit from the switch).

Symbols: Extending (in a natural way) the names given in Section 2, Q_{ij} is the queue from input i to output j in the mesh of tree, and L_{ij} is its length.

In order to establish that the mesh of trees switch mimics an OQ switch (with a delay, as we will see), it is useful to introduce an intermediate architecture that will be used as a paragon. In Fig. 4, it is shown a queued architecture for a single output (referred in the remainder of this section as DFIFO[3]) composed by n queues, from which the K element chooses the oldest to forward (breaking ties in the usual way), and $\log_2(n)-1$ elements that just forward from one end to the other (actually the architecture is just the one shown in Fig. 2, with $\log_2(n)-1$ more stages).

[3] DFIFO is just a short name for Delayed FIFO.

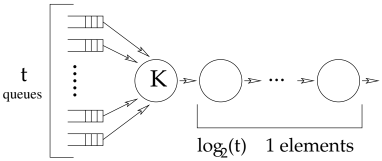


Fig. 4. An intermediate architecture.

It is now useful to introduce some straightforward lemmas:

Lemma 1. *A switch with DFIFO queuing architecture mimics a FCFS* OQ *switch with a delay of* $\log_2(n)$ *steps.*

Proof. As noted in Sect. 3, the architecture that, for every output, selects the oldest cell from t queues, exactly behaves like a FIFO OQ switch. Adding $\log_2(n)$ forwarding units, we just introduce a delay in the output.□

Lemma 2. *A mesh of trees switch mimics a switch with DFIFO queuing architecture with a delay of* $\log_2(n)$ *steps.*

Proof. The delay is caused by the selection phase done at the column trees: as we assume infinite size queues, it takes exactly $\log_2(n)$ time slots to a cell for arriving to the queues, while in a DFIFO based switch they would arrive in one step. So it is enough to show that, without considering the column trees in the mesh of trees, the two architectures are totally equivalent.
We focus on an output j, in order to prove that cells bound for that output are handled in the same way (once they arrive at the queues) by the two architectures. Since we don't make any assumption on j, this will hold for all the outputs, establishing the lemma. We shall prove the lemma by induction on the number of queues[4]:

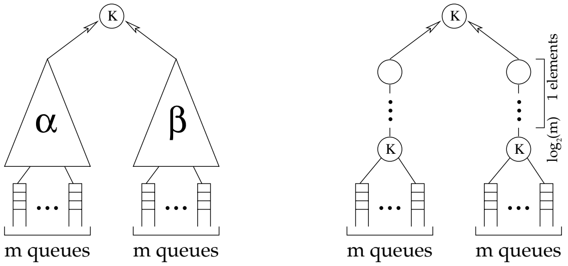
basic step: for $n = 2$ (2 inputs/outputs, the minimum case), the two architectures are exactly the same (the K element that chooses between two queues),

induction step: for $n = 2m$ and $m > 1$ the row tree can be seen as the composition of two trees with m leaves (queues) (see Fig. 5(a)). By induction, such an architecture is equivalent to the one shown in Fig. 5(b).

It is not hard to show the equivalence of such an architecture and a DFIFO of height $\log_2(m)$ (or equivalently $\log_2(2m) - 1$) forwarding elements. To avoid tedious details, it can be sufficient noting that

- the number of steps that cells must undergo, is the same ($\log_2(2m)$ after the first selection),
- during any time slot, if at level i of Fig. 5(b) architecture there is one cell, then there is one cell also at the same level of the DFIFO architecture,

[4] Given the mesh of trees features, we only deal with powers of 2, with 2 as bottom of the induction chain.



(a) Trees composition

(b) DFIFO composition

Fig. 5. Inductive step.

- in any time slot, if at level i of Fig. 5(b) architecture there are two cells, then in the DFIFO architecture there is one cell at level i and one cell at level $i + 1$,
- inversely, at any time slot, if at level i of DFIFO architecture there is one cell, then either there is at least one cell at the same level of Fig. 5(b) architecture, or there are two cells at level $i - 1$ (note that this holds for $i > 0$ since the root is unique in both systems).

So, at every time slot there is a cell in output in one architecture if and only if there is a cell in output in the other. Since outputs are time ordered, they must be exactly the same.□

Lemma 3. *Consider three switch architectures A, B and C. If A mimics B with a delay of $f(n)$ and B mimics C with a delay of $g(n)$, then A mimics C with a delay of $f(n) + g(n)$.*

Proof. By definition of mimicking with a delay (see Sect. 2), under the same arrivals, the output of C at time t is the same of B at time $t + g(n)$, which in turn is the same of A at time $t + g(n) + f(n)$.□

We have thus established the following

Theorem 1. *The mesh of trees switch mimics a FCFS* OQ *switch with a delay of* $2\log_2(n)$.

□

The mesh of trees architecture is particularly suitable to efficiently provide multicast. An addressing technique already known suffices: the destination of every cell is coded by a t bits string with the i^{th} bit set to 1 if and only if the output i is in the set of receivers. So, during the selection stage, the node at level

i in the column tree must only perform two "or" operations when a cell to route arrives: one of the bits in the left half of the word, and one of the rightmost ones. If the first "or" operation is equal to 1, then the cell is forwarded to the left child (with the left half word as destination information), and the same happens with the second operation, but the cell is forwarded to the right child (note that at least one operation must be positive, but both can produce a 1). The so implemented multicast is a copy multicast, and it is the most efficient way to implement it: the cells arrive at the queues during the same time slot (because of the synchronism of the selection phase), and will depart during the first empty time slot.

We believe this feature makes particularly interesting the proposed switch: the IQ architecture in fact has several problems managing multicast traffic, both from a theoretical point of view ([36]) and from a practical one (e.g. the simulations results in [37]), while in the mesh of trees switch the scheduling of multicast traffic practically comes for free. As previously established, the mesh of trees switch can mimic a FCFS OQ switch. Besides, for very large n's the OQ switch can be considered purely theoretical because of the needed speedup, while the mesh of trees scales very well. Moreover, the time slot length limit is given just by the memory speed: in fact, the whole architecture behaves like a pipeline, and the time of the system is given by the time of the slowest element. If a comparing step is faster than a memory cycle, we can think to group several comparing steps into a single system cycle, in order to reduce the delay of the mesh of trees and to improve performances.

The mesh of trees architecture seems to suffer of the same spatial speedup problem of the knockout switch: the queues in the leaves, for graphical presentation reasons, are drawn as adjacent, and at a first sight they can be imagined as a single big memory with a speedup problem. In the real physical implementation, memories not necessarily are positioned as in Fig. 3(a). Moreover, we think that, at least theoretically, the study of such a kind of architectures can be interesting, because of the positive performances offered that can overcome technical problems.

6 Conclusions

In this paper, we considered a parallel architecture for the implementation of the well known output queued switch. The widely studied mesh of trees topology has been used to propose a switch that can mimic (even if with a logarithmic delay) a FCFS output queued switch without the speedup problem. A future work will be to extend the class of queuing policies that is possible to emulate, in order to achieve quality of service, and to give some bounds on queues sizes and dimension of time stamps needed.

References

[1] M. G. Hluchyj and M. J. Karol. Queueing in high-performance packet switching. *IEEE Journal on Selected Areas in Communications*, 6(9):1587–1597, Dec. 1988.

[2] T. E. Anderson, S. S. Owicki, J. B. Saxe, and C. P. Thacker. High-speed switch scheduling for local-area networks. *ACM Transactions on Computer Systems*, 11(4):319–352, Nov. 1993.

[3] N. Mckeown, M. Izzard, A. Mekkittikul, W. Ellersick, and M. Horowitz. The tiny tera: a packet core switch. *Hot Interconnects IV, (Sstanford University)*, pages 161–173, Aug. 1996.

[4] C. Partridge, P. P. Carvey, E. Burgess, I. Castineyra, T. Clarke, L. Graham, M. Hathaway, P. Herman, A. King, S. Kohalmi, T. Ma, J. Mcallen, T. Mendez, W. C. Milliken, R. Pettyjohn, J. Rokosz, J. Seeger, M. Sollins, S. Storch, B. Tober, G. D. Troxel, D. Waitzman, and S. Winterble. A 50 gb/s ip router. *IEEE/ACM Transactions on Networking*, 6(3):237–248, Jun. 1998.

[5] J. P. Coudreuse and M. Servel. PRELUDE: an asynchronous time-division switched network. In *Proceedings of IEEE International Conference on Communications '87*, pages 769–773, 1987.

[6] N. Endo, T. Kozaki, T. Ohuchi, H. Kuwahara, and S. Gohara. Shared buffer memory switch for an ATM exchange. *IEEE Transactions on Communications*, 41(1):237–245, Jan. 1993.

[7] M. J. Karol, M. G. Hluchyj, and S. Morgan. Input versus output queueing on a space division switch. *IEEE Transactions on Communications*, 35:1347–1356, 1987.

[8] H. Zhang. Service disciplines for guaranteed performance service in packet switching networks. *Proceedings of the IEEE*, 83(10):1374–1396, Oct 1995.

[9] M. Karol, K. Eng, and H. Obara. Improving the performance of input-queued atm packet-switching. In *Proceedings of IEEE INFOCOM '92*, pages 110–115, 1992.

[10] L. Tassiulas and A. Ephremides. Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. *IEEE Transactions on Automatic Control*, 37(12):1936–1948, Dec. 1992.

[11] L. Tassiulas. Linear complexity algorithms for maximum throughput in radio networks and input queued switches. In *Proceedings of IEEE INFOCOM '98*, pages 533–539, 1998.

[12] N. McKeown, V. Anantharam, and J. Walrand. Achieving 100% throughput in an input-queued switch. In *Proceedings of IEEE INFOCOM '96*, pages 296–302, 1996.

[13] Y. Li, S. Panwar, and H. J. Chao. On the performance of a dual round-robin switch. In *Proc. of IEEE Infocom 2001*, 2001.

[14] A. Mekkittikul and N. McKeown. A starvation-free algorithm for achieving 100% throughput in an input- queued switch. In *Proceedings of the ICCCN*, pages 226–231, 1996.

[15] N. McKeown. *Scheduling algorithms for input queued cell switches.* PhD thesis, University of California at Berkeley, 1995.

[16] C.S. Chang, W.J. Chen, and H.Y. Huang. On service guarantees for input buffered crossbar switches: a capacity decomposition approach by birkoff and von neumann. In *IEEE IWQoS'99*, pages 79–86, 1999.

[17] R. E. Tarjan. *Data structures and network algorithms.* Society for industrial and apllied mathematics, 1983.

[18] C.S. Chang, W.J. Chen, and H.Y. Huang. Birkhoff-von neumann input buffered crossbar switches. In *Proc. of IEEE Infocom 2000*, 2000.

[19] N. McKeown. The islip scheduling algorithm for input-queued switches. *IEEE/ ACM Transactions on Networking*, 7(2):188–201, Apr. 1999.

[20] M. W. Goudreau, S. G. Kolliopoulos, and S. B. Rao. Scheduling algorithms for input-queued switches: randomized techniques and experimental evaluation. In *Proc. of IEEE Infocom 2000*, 2000.

[21] W. Leland, M. Taqqu, W. Willinger, and D. Wilson. On the self-similar nature of ethernet traffic (extended version, 1994.

[22] E. Leonardi, M. Mellia, F. Neri, and M. Ajmone Marsan. Bounds on average delays and queue size averages and variances in input-queued cell based switches. In *Proc. of IEEE Infocom 2001*, 2001.

[23] S. T. Chuang, A. Goel, N. McKeown, and B. Prabhakar. Matching output queueing with a combined input output queued switch. *IEEE Journal on Selected Areas in Communications*, 17(6):1030–1039, 1999. (A preliminary version appears in Proceedings of INFOCOM '99).

[24] B. Prabhakar and N. McKeown. On the speedup required for conbined input and output queued switching. *Automatica*, 35(12):1909–1920, Dec. 1999.

[25] M. Andrews and L. Zhang. Achieving stability in networks of input-queued switches. In *Proc. of IEEE Infocom 2001*, 2001.

[26] F. M. Chiussi, D. A. Khotimsky, and S. Krihsnan. Generalized inverse multiplexing of switched atm connections. In *Proc. of IEEE Globecom '98*, 1998.

[27] F. M. Chiussi, D. A. Khotimsky, and S. Krihsnan. Advanced frame recovery in switched connection inverse multiplexing for atm. In *Proc. of IEEE International Conference on ATM '99*, 1999.

[28] D. A. Khotimsky and S. Krihsnan. Stability analysis of a parallel packet switch with bufferless input demultiplexor. In *Proc. of IEEE ICC 2001*, 2001.

[29] S. Iyer, A. Awadallah, and N. McKeown. Analysis of a packet switch with memories running slower than the line-rate. In *Proceedings of IEEE INFOCOM 2000*, 2000.

[30] S. Iyer and N. McKeown. Making parallel packet switches practical. In *Proceedings of IEEE INFOCOM 2001*, 2001.

[31] A. Mekkitikul and N. McKeown. A practical scheduling algorithm to achieve 100% throughput in input-queued switches. In *Proceedings of IEEE INFOCOM '98*, pages 792–799, 1998.

[32] A. Mekkittikul. *Scheduling non-uniform traffic in high speed packet switches and routers*. PhD thesis, Stanford University, 1998.

[33] Y. S. Yeh, M. G. Hluchyj, and A. S. Acampora. The knockout switch: A simple modular architecture for high performance switching. *IEEE Journal on Selected Areas in Communications*, SAC-5:1274–1283, Oct. 1987.

[34] F. T. Leighton. *Introduction to parallel algorithms and architectures: arrays, trees, h ypercubes*. Morgan Kaufmann, 1992.

[35] F. P. Preparata and J. E. Vuillemin. Area-time optimal vlsi networks for matrix multiplication. 11(2):77–80, 1980.

[36] Z. Liu and R. Righter. Scheduling multicast input-queued switches. *Journal of scheduling*, 2(3):99–114, May 1999.

[37] M. Ajmone Marsan, A. Bianco, P. Giaccone, E. Leonardi, and F. Neri. On the throughput of input-queued cell-based switches with multicast traffic. In *Proc. of IEEE Infocom 2001*, 2001.

Fault-Tolerant Support for Reliable Multicast in Mobile Wireless Systems

Giuseppe Anastasi[1], Alberto Bartoli[2], and Flaminia L. Luccio[3]

[1] Dip. di Ingegneria dell'Informazione, Università di Pisa, Italy
anastasi@iet.unipi.it
[2] Dip. di Elettrotecnica, Elettronica e Informatica Università di Trieste, Italy
bartolia@univ.trieste.it
[3] Dip. di Scienze Matematiche, Università di Trieste, Italy
luccio@dsm.univ.trieste.it

Abstract. In this paper we present a protocol for reliable multicast within a group of mobile hosts that communicate with a wired infrastructure by means of wireless technology. The protocol tolerates failures in the wired infrastructure, i.e., crashes of stationary hosts and partitions of wired links. The wireless coverage may be incomplete and message losses could occur even within cells, due to physical obstructions or to the high error rate of the wireless technology, for example. Movements of mobile hosts are accommodated efficiently because they do not trigger any interaction among stationary hosts (i.e., there is no notion of hand-off). We evaluate by simulation the impact of fault-tolerance on the performance of the protocol in normal operating conditions, i.e., in the absence of failures. The results obtained show that the increase in the average latency experienced by messages is limited to few milliseconds.

1 Introduction

Computing architectures based on *portable computers* and *wireless networking* are becoming a reality. Users may be equipped with hand-held computing devices and roam around freely while maintaining connectivity with a wired computing infrastructure through a number of wireless cells.

Mobile wireless systems typically require special solutions, for a number of reasons. Traditional network protocols implicitly assume that hosts do not change their physical location over time. Mobile devices have severe resource constraints in terms of energy, processing and storage resources. Wireless networks are characterized by limited bandwidths and high error rates. Furthermore, mobility introduces new issues at the algorithmic level. For example, a mobile host may miss messages simply because of its movements, even with perfectly reliable communication links and computers that never crash [1]. All the above reasons imply that specialized protocols are required for extending to mobile hosts functionalities common for stationary ones.

In this paper we present a protocol for *reliable* and *totally-ordered* multicast within a group of mobile hosts. By this we mean that: (i) each mobile host delivers all multicasts, without duplicates; and (ii) any two mobile hosts that deliver two multicasts deliver these multicasts in the same order.

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 1069–1080, 2002.

Reliable and totally-ordered multicast is an important building block for applications composed of remote processes that have to cooperate tightly [13]. This communication primitive has proven its power in the context of traditional, i.e., static and wired, distributed computing. Our proposal makes this primitive available on mobile wireless systems. Moreover, we support this primitive in spite of (a certain number of) crashes of stationary hosts and partitions of wired links. The fault-tolerance properties of our protocol may greatly extend the scope of potential applications of mobile computing, including emergency management, plant control, traffic monitoring, stock market exchange, on-site data collection, for example. Fault-tolerant support for mobile wireless systems is, in our opinion, an important topic, yet it has not received much attention from the research community so far.

We model a mobile wireless system as follows (see figure 1). There is a set of *stationary hosts* (SHs) connected by a wired network and a set of *mobile hosts* (MHs) that may move and communicate through wireless links. Some SHs, called *mobile support stations* (MSSs), may communicate also through wireless links. Each MSS defines a spatially limited *cell* covered by a wireless link. A MSS may broadcast messages to all MHs in its cell and send messages to a specific MH in its cell, whereas a MH may only send messages to the MSS of the cell where it happens to be located. Notice that we do not assume any network support for routing messages to a specific MH.

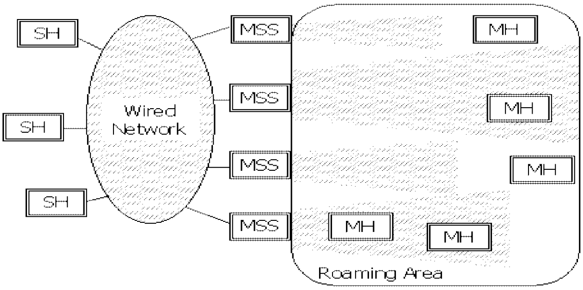


Fig. 1. Example system with five MHs and seven SHs.

An important feature of our model is the *incomplete coverage* of wireless cells, i.e., MHs may roam in areas that are not covered by any cell. A MH may move across adjacent cells but it may also "disappear" within the uncovered area and enter any other cell, perhaps after a "long" time. Movements occur without prior negotiation. The resulting scenario is quite general because it accommodates contemporary wireless LAN's, infra-red networks requiring line-of-sight connectivity, disconnected modes of operation, long-range movements and picocellular wireless networks in which the cell size is of the order of a few meters, such as a room in a building.

The message pattern of our protocol follows common approaches for reliable multicasting among MHs in mobile wireless systems [1,3,8,14,16]. A MH wishing to issue a multicast sends a request to the MSS of the cell where it happens to be located. The MSS forwards the message to a SH that processes this request, includes the payload, and forwards the payload to all MSSs. MSSs broadcast the payload in the respective cell. More details will be given later.

Our work is based on a design philosophy aimed to improve reliability of final applications, in particular, with respect to failures:

1. The state shared among SHs should not be updated upon *each* movement of MHs. Otherwise, performance could be penalized and failure handling would be more difficult.
2. One should avoid to assume that wireless coverage is complete. Otherwise, even a single physical obstruction, or particularly unfortunate area, or MSS malfunctioning, could compromise correctness.
3. Availability of MSSs should affect only availability of applications, not their correctness. In particular, a MSS failure should merely shrink the covered area, without affecting correctness.
4. One should avoid to make hypothesis on users' movements. Otherwise, even a *single* inopportune movement could compromise correctness.
5. Critical state information should not be kept on MSSs, but on "ordinary" SHs. This choice allows using systematic and established techniques for improving the availability of these hosts, such as replication.
6. MSSs should be freely added or removed without stopping the system or compromising correctness. MSS addition may be necessary for upgrading or coverage enhancement, whereas MSS removal for maintenance or failure.

Notice that the above points apply to mobile computing in general, not only to the specific problem of reliable multicast.

We have analyzed the performance of the proposed protocol by simulation. In particular, we have focused on the impact of fault-tolerance on the performance of the protocol in normal operating conditions, i.e., in the absence of failures. We have found that the proposal is indeed practical, as the latency increase due to fault-tolerance is of just a few milliseconds.

2 System Model

Each wired link and each wireless cell provides FIFO-ordered communication without duplicates. Messages may be lost. Message loss in a wired link occurs as a result of network partitions. Such partitions may recover. Message loss in a wireless cell may occur because of physical obstructions or because of the intrinsic features of wireless technology, e.g., high error rate. Hosts communicate solely via messages. Of course, while a MH is out of coverage no communication with it is possible. Similarly, SHs partitioned from each other cannot communicate among themselves. SHs may crash and a crashed SH may recover. MHs do not crash (see also below).

The system is asynchronous in the sense that neither message delays nor computing speeds can be bounded with certainty. This characterization is a general and realistic one as it allows abstracting away such features as variable loads imposed by users and unknown scheduling strategies on hosts and communication links. Notice that a process cannot determine with certainty whether a remote process that appears to be unresponsive has crashed or happens to be very slow.

The protocol can be easily made resilient also to crashes of MHs, the only problem being that state information about a crashed MH would never be discarded by SHs. A practical implementation might allow SHs to unilaterally garbage-collect state information not accessed for a very long time. Although a MH deemed crashed might show up again, there is *no* way to exclude such a possibility in an asynchronous system — unless one is willing to wait for an infinite time before deciding whether that MH actually crashed. Another practical issue is that a crashed MH should be able to participate again in the application after its recovery. This feature may be achieved by: (i) supporting a *dynamic* group of MHs that may exchange multicasts; and (ii) requiring that multicasts be delivered only by current members of the group. The protocol proposed here assumes a static set of MHs but it may be extended towards supporting (i) and (ii) quite simply [6, 9].

3 Related Work

To the best of our knowledge, the only work with scope similar to ours is [2]. This work introduces resilience to failures of MSSs in a non fault-tolerant reliable multicast protocol proposed by Acharya and Badrinath (see below, [1]). However the system model is much more restrictive than ours because: (i) it assumes that a process can detect with *certainty* whether a remote process is active or crashed (fail-stop failures); and (ii) communication is reliable, both in the wired network and in wireless cells (thereby excluding, for example, uncovered regions, physical obstructions within cells, partitions of wired links).

The protocol in [2] adds fault-tolerance to the one in [1] by associating each MH with a set of MSSs, denoted $\mathcal{S}$(MH), and by replicating state information about that MH at each member of $\mathcal{S}$(MH). Whenever MH sends a message or there is a message addressed to it, members of $\mathcal{S}$(MH) have to execute a replica control protocol and this protocol must be able to tolerate host failures. The cited work mentions two alternatives for such protocol. The one that is more efficient requires additional mechanisms (network flush or rollback) that are not detailed. Furthermore, no performance analysis is provided and the complex interaction among (i) replica control protocol, (ii) MSS recovery, and (iii) hand-off, are only outlined [1]. Our protocol is fully detailed and, in our opinion, is much simpler to understand and implement.

The protocol by Acharya and Badrinath, hereinafter the AB-protocol, was the first multicast protocol ensuring reliable (FIFO) delivery in the context of mobile computing and has been highly influential in the design of later protocols [3,8,14,16]. Although none of these protocols is fault-tolerant, it is useful to discuss them briefly to emphasize the differences with our proposal. Each MSS maintains, for each MH in its cell, an array of sequence numbers describing the multicasts already delivered by that MH. The MSS uses this array to forward pending messages in sequence and without duplicates. If the MH switches cell, the array is moved to the new MSS by means of a proper *hand-off*

[1] The paper claims that the composition of $\mathcal{S}$(MH) may change dynamically, but it appears that this issue has been oversimplified, in particular, with respect to the interaction just mentioned.

procedure. Therefore: (i) The state shared among SHs is updated upon *each* movement; (ii) MSSs maintain critical state information (i.e., each MSS remembers the sequence numbers of multicasts delivered by each MH in its cell); and (iii) the crash of a MSS affects correctness of the application (i.e., the above sequence numbers are lost for each MH in the cell). These features explain why the fault-tolerant extension in [2] requires a complex interaction among several sub-protocols. The AB-protocol and the protocols derived from it assume reliable communication in the wired network and in the wireless network, much like [2].

The AB-protocol provides reliable delivery without requiring routing support for MHs, e.g., Mobile IP, much like our proposal. Multicast protocols that rely on Mobile IP are generally targeted at different application domains and provide unreliable, best-effort, unsequenced delivery [12,15]. In particular, no messages are delivered during a cell switching and messages possibly lost will not be recovered in the new cell. With respect to the use of Mobile IP, note also that: (i) it would not solve the problem of recovering from lost messages; (ii) it would make it more difficult to exploit the broadcast capabilities of the wireless medium when many MHs are in the same cell; (iii) it would generate traffic in the wired network even while no new multicasts are generated, for tracking the location of each MH.

The protocol proposed here is an extension of the protocol in [9] that was not fault-tolerant and assumed reliable communication in the wired network. As an aside, performance analysis by simulation showed that the proposal in [9] outperforms the AB-protocol in terms of latency, scalability, bandwidth usage efficiency and quickness in managing cell switches of users [4]. The proposal in [9], as well as the one in this paper, borrows a crucial idea from the implementation of reliable multicasts in "static and wired" distributed systems: the use of a centralized sequencer for totally ordering multicasts and for storing multicasts that have not been acknowledged yet [13]. Note, however, that here we refer to a completely different system model: mobile hosts, wireless communication, incomplete spatial coverage.

4 Overview of the Protocol

We begin by briefly outlining the non fault-tolerant version of the protocol. Messages have a field of enumerated type, called tag and indicated in SMALLCAPS, that indicates the purpose of the message. We say that a host H *receives* a message m when m arrives at the protocol layer at H, and that H *delivers* m when the protocol forwards m up to the application.

A MH wishing to issue a multicast sends the payload to the local MSS with a NEW message. MH retransmits this message until receiving an acknowledgment (possibly from a different MSS, if the sending MH moves during the handshake). The MSS forwards the message to a designated SH acting as *coordinator*. A NEW message carries a sequence number locally generated by the sending MH, which enables the coordinator to process NEW messages in sequence and to discard duplicates. The coordinator constructs a NORMAL message containing the payload of the NEW message and a locally generated sequence number. The resulting message is then multicast to MSSs that broadcast it in the respective cell. Each MH uses sequence numbers of NORMAL messages to deliver these messages in sequence without duplicates (i.e., in total order) and to detect missing messages. In the latter case, the MH sends a retransmission request to the local MSS. This request is tagged NACK and specifies an interval of missing sequence numbers. When a MSS receives a NACK, it relays the missing NORMAL messages to the sending MH. The MSS obtains such messages from a local *cache* or, in case of a miss, from the coordinator. MSS requests missing messages to the coordinator with a FETCHREQ specifying an interval of sequence numbers. The coordinator responds with a FETCHREP containing the required messages. A NACK from a MH implicitly acknowledges delivery of previous multicasts. MSSs extract this information and forward it to the coordinator, with STABINFO messages. Note that: (i) MSSs do not store critical state information: such information is kept by the coordinator and merely cached by MSSs for efficiency; (ii) each MSS reacts to cell switching without interacting with other MSSs.

The *fault-tolerant* extension proposed here is obtained as follows.

1. A MH no longer assumes that a message arrived at a MSS will eventually arrive at the coordinator — the MSS might crash, or a partition might occur. Instead, a MH keeps on retransmitting a NEW message until receiving the matching NORMAL message (a MSS that receives a NEW message does not respond to the sending MH with an acknowledgment, as this acknowledgment would be useless).
2. The role of the single coordinator is played by *a set* of SHs, called *coordinators*. This set appears to MSSs as a single "coordinator service". The service is available in spite of (a certain number of) failures of coordinators and connecting links. In particular, availability of the service requires a majority of coordinators. Coordinators interact among themselves through *group communication (GC)* [10]. GC may be thought of as a software layer exporting to applications a membership service and a communication service for reliable multicasting within a group of processes. These two services are tightly integrated so as to simplify the programming of distributed algorithms in the face of host crashes and recoveries, network partitions and mergers. More details will be given in section 4.1.
3. A MSS sends its messages to a designated coordinator, say C. The MSS might not receive a response for several reasons, including: (i) C is not able to interact with a majority of coordinators (section 4.1); (ii) the message from MSS to C is lost; (iii) the response from C to MSS is lost. Should a response not arrive within a specified timeout, the MSS will send the *next* request to another coordinator. The request not yet answered will be retransmitted by the originating MH, as pointed out above (1). The policy for associating coordinators with MSSs is irrelevant to this paper. Of course, timeouts expiring too soon must not affect correctness. To this end, the coordinator service maintains internally information sufficient to detect duplicate requests (section 4.1).

Space constraints preclude a full description of the protocol, that can be found in the companion report in a pseudo-code form [5]. We will discuss in the next section only the implementation of the coordinator service.

4.1 Coordinator Service

Interaction among coordinators occurs through *group communication (GC)* [10]. A detailed description of GC is beyond the scope of this paper and we provide below only the necessary background. GC is implemented by a dedicated software layer at each coordinator.

Coordinators form a *group* (this notion of group has nothing to do with the group of MHs). The GC layer provides consistent information about the set of coordinators that appear to be currently reachable. This information takes the form of *views*. The GC layer determines a new view as a result of crashes, recoveries, network partitions and mergers. New views are communicated to coordinators automatically, through special messages called *view changes*. When a coordinator C receives a view change carrying the new view V, C is informed that it can communicate with the coordinators listed in V. To proceed further, we need a few simple definitions: (i) C *installed* a view V means that C indeed received the corresponding view change; (ii) two views V, W are *consecutive* means that a coordinator installs V and then installs W; (iii) the view that is *current* at C is the one specified by the last view change received by C; (iv) C delivers message m *in view* V means that C delivers m when the view that is current at C is V.

The key guarantee of GC is that view changes are globally ordered with respect to the receiving of multicasts: *Given two consecutive views* V *and* W*, any two coordinators that install both views must have received the same set of multicast messages in view* V. For example, consider a coordinator C_1 that delivered V and suppose W is delivered as a result of the crash of C_1. If C_1 crashed while performing a multicast m, then (i) all coordinators that install V and W receive m (and do so *before* installing W); or (ii) none of them receives m. Clearly, this property is very powerful for reasoning about fault-tolerant algorithms.

The GC layer supports *partitionable* membership, i.e., it allows multiple views of the group to exist concurrently, to model network partitions. Moreover, the GC layer supports *uniform multicast*: if any member of view V delivers multicast m, then each member of V delivers m or crashes. We present the algorithm in the hypothesis that a view including a majority of coordinators always exists. The algorithm may be extended to accommodate the more general case in which the majority view temporarily disappears.

The variables maintained by each coordinator include the following: `boss`, the identifier of a designated member of a majority view; `cseq`, the sequence number of the last NORMAL message sent; `normal-buffer`, a set containing all NORMAL messages that might not be *stable*, i.e., that are not known to have been delivered by each MH; finally, `member-table`, a table with one element for each MH. Each element is a record whose fields are: `mid`, that identifies the MH; `new-num`, the sequence number (generated by the MH) of the last NEW message received from `mid`; `cseq-mid`, the `cseq` assigned to the last NORMAL message generated upon processing a NEW sent by `mid`; `delivered`, the highest sequence number of a NORMAL message that has certainly been delivered by `mid`.

Each coordinator C executes a loop in which at each iteration it receives either a message or a view change. If the current view is not a majority, C skips to the next iteration — C ignores all messages and waits for a sufficient number of failures to recover. Otherwise, C acts as follows. Receiving of a message provokes the transmission of an ACK to the sending MSS (to prevent expiration of the time-out at MSS). In addition:

- A NEW message is forwarded to the `boss`. When the `boss` receives one such message, it multicasts the message within the majority view. Let m denote a message multicast by the boss and let mid denote the MH that originated the associated New message. Upon receiving m, each coordinator C performs the following actions: (i) extract the entry, say e-mid, of member-table associated with mid; (ii) determine whether m is a duplicate and, in this case, discard m without any further processing (this check is done by comparing field new-num of e-mid to the sequence number in m, selected by mid itself); (iii) update field new-num of e-mid; (iv) increase cseq (system-wide sequence number); (v) construct a Normal message m_N including, in particular, the payload specified by mid and cseq; (vi) store a copy of m_N in normal-buffer; finally, (vii) the boss multicasts m_N to MSSs. In short, coordinators proceed in locksteps and, in particular, they maintain identical copies of their variables.

- A FetchReq message is processed locally (such a message is sent by a MSS whose local cache does not contain a Normal message requested by a MH). The FetchRep reply is constructed based on the normal-buffer.
- A StabInfo message is multicast within the view (such a message describes the Normal messages certainly delivered by a specified MH). Upon receiving this multicast, each coordinator records the related information in the pertinent entry of member-table and clears from normal-buffer messages that have been delivered by every group member.

Network partitions, mergers, host crashes and recoveries are handled simply. GC reports them automatically to coordinators in the form of a new view. Upon receiving a view change, the boss sends a copy of its variables to each coordinator that was not in the previous majority view. Then, the coordinator service starts processing again messages from MSSs, as all coordinators in the new majority view have identical variables. If the boss has left the majority view (e.g., it crashed), then a new boss is elected by applying a deterministic function to the composition of the view (this function must select a member of the previous majority view). Variables to the coordinators that have possibly entered the majority view will be sent by the new boss. Notice that all coordinators receive the same view, hence they can easily coordinate their reaction to the view change based solely on the view composition, i.e., without dedicated message-exchange rounds.

It may be useful to observe what follows: (1) The boss might crash during steps (i)-(vii) above, i.e., before actually multicasting the Normal message to MSSs. In this case, MHs will eventually detect a hole in the stream of sequence numbers and ask retransmission; (2) When surviving coordinators receive the view notifying about the crash of the boss, they will certainly have the same variables: GC ensures that prior to the view change they have delivered the *same* set of multicasts from the boss.

5 Simulation

Fault-tolerance obviously comes at a cost. A protocol designed to be fault-tolerant is likely to exhibit, *even in the absence of failures*, performance worse than that of a protocol that does not tolerate failures. In this section we evaluate such costs by simulation. This analysis enables us to capture the inherent cost of fault-tolerance for our proposal. Accordingly, our simulations assume reliable communication in the wired network and SHs that do not crash. The emphasis here is demonstrating that one can tolerate failures without paying excessive costs in normal conditions, i.e., in the absence of failures.

We set the numerous parameters that characterize the protocol similar to [6], which provides a simulation analysis for the non fault-tolerant version. There are 40 cells, i.e., 40 MSSs. A MH remains in a cell for a random time interval. The length of this interval is exponentially distributed and its average T_{cell} is set to 10 seconds for each MH. Wireless coverage is complete and the message loss rate in the wireless network is 0.1%. There are 100 MHs: all of them receive multicasts (N_r=100) whereas only 10 of them may generate 512-byte messages (N_s=10). Message generation is a Poisson process, i.e., times between the generation of successive messages are random variables exponentially distributed. Each sender generates, on the average, 8 messages/sec corresponding to a bit rate of approximately 33 Kbps.

We consider a wireless bandwidth of 1 Mbps, in line with the bandwidth available in current Wireless LANs [7], and a wired bandwidth of 10 Mbps. Therefore, message transmission times in the wired network are one order of magnitude lower. Propagation delays, i.e., times messages take to travel from one node to another, are as follows. Wireless propagation delays are negligible as cells are supposed to be very small, (e.g., ten meters). Wired propagation delays are assumed to be exponentially distributed. The average for messages from MSSs to coordinators or back is 1.5 msec, whereas for messages from coordinators to the boss is 2 msec (these point-to-point messages are sent through group communication). Uniform multicast amongst coordinators is modeled as an additional exponential delay with average $(N_c + 0.4)$ msec ([11]). We consider also processing time, i.e., time necessary to process a message. We used the same values reported in [6] as there are no substantial differences from the non fault-tolerant version.

The main metric we consider is the *average message latency*, i.e., the average time elapsed from the instant at which a message is generated at a sending MH to the instant at which the same message is delivered by a destination MH. We analyzed latency for varying numbers of coordinators, sending MHs, mobility of MHs and message loss rate. Curves labelled as $N_c = 1$ refer to the non fault-tolerant version while the other curves relate to the fault-tolerant protocol proposed in this paper. For the sake of space we shall focus on the differences between the two versions. The reader can refer to [6] for details about the performance of the non fault-tolerant version.

Figure 2-left shows the average latency as a function of the number of receivers for different number of coordinators (N_c). Note that the fault-tolerant version maintains the very good scalability properties of the non fault-tolerant one. The fault-tolerant protocol exhibits higher average latency as a result of the following factors:

1. Messages experience an additional step with respect to the non fault-tolerant version: from the coordinator associated with the MH that originated the message to the boss.
2. When the boss receives a NEW message it does not multicast the related NORMAL message immediately, but it sends a uniform multicast within the view and waits for delivery of this multicast.
3. NEW messages are implicitly acknowledged by the related NORMAL message from the boss while in the non fault-tolerant version they are explicitly acknowledged by the local MSS. It follows that MHs have to use longer time-outs in order to minimize useless retransmissions, but this delays retransmission of messages that are actually lost.

The companion report [5] analyzes in more detail the contribution of each factor.

Figure 2-right shows the average latency as a function of the number of senders, i.e., of the aggregate message rate. Although average latency increases with the number of senders, it is important to observe that curves for different values of N_c are approximately parallel. In other words, the fault-tolerant version maintains approximately the same scalability properties of the non fault-tolerant one. The difference between the two protocols is due, obviously, to points 1, 2 and 3 above.

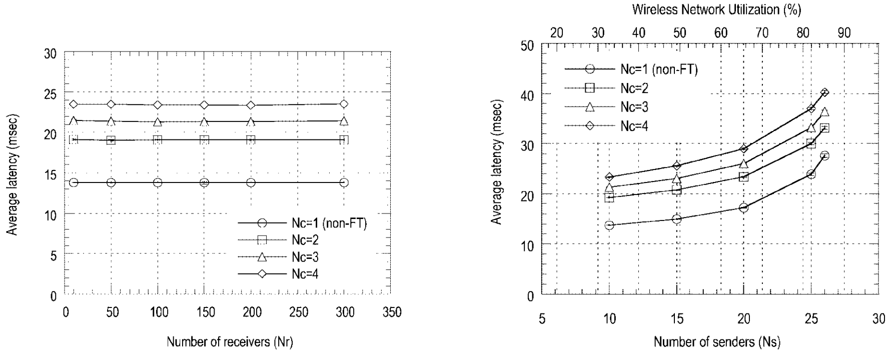


Fig. 2. Average latency as a function of the number of receivers (left) and of the number of senders (right) for different number of coordinators.

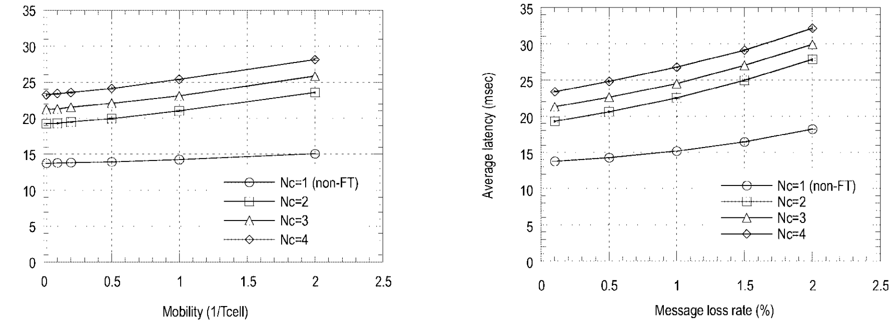


Fig. 3. Average latency as a function of mobility (left) and wireless network unreliability (right) for different number of coordinators.

Figure 3 shows the influence of MH mobility (left) and wireless network unreliability (right). Mobility is expressed in terms of the number of cell switches per second experienced by each MH, which is the inverse of the T_{cell} parameter, i.e., the average cell permanence time. Unreliability of wireless links is expressed as the percentage of lost messages.

Both plots exhibit a similar behavior. In particular, the difference between the non fault-tolerant protocol and the fault-tolerant protocol (for example with N_c=2) increases as either mobility or message loss rate grows up. This similarity can be easily understood if one considers that mobility of MHs may cause message losses. The increase in the distance between curves related to $N_c = 1$ and $N_c > 1$, respectively, is a consequence of point 3 above: when the fraction of NEW messages which get lost increases, the delay for recovering them increases accordingly.

To summarize, the latency cost induced by fault-tolerance in the absence of failures is in the order of a few msec. The above components 1 and 2 of the additional delay cannot be reduced (for a fixed wired network technology and operating environment). On the other hand, component 3 could be partially lowered by using at MHs:(i) a transmission scheme more sophisticated than the simple stop and wait approach (e.g., a window-based scheme); and/or (ii) a shorter time-out for NEW messages. On the other hand, the former would induce higher computational load at MHs while the latter would cause useless retransmissions and, thus, wastage of wireless bandwidth as well as computing and energy resources at the MH. Based on the above results, we believe that these solutions would lead to minor performance improvements that would not compensate for their drawbacks. However, in a different scenario, e.g., when MSSs are distributed in a geographical area rather than in a local area, use of a window-based transmission scheme could be appealing.

6 Concluding Remarks

We have presented a protocol for offering fault-tolerant support to (totally ordered) reliable multicast within a group of MHs. The protocol tolerates crashes of SHs and partitions of wired links. To our knowledge, no other protocol provides these functionalities.

Two key features of our protocol are: (i) movements of MHs do not require any interaction among SHs (i.e., no hand-off is required); and (ii) MSSs do not store any critical state information. Both features are crucial for coping with failures simply and efficiently. MSSs merely act as forwarding switch and as *cache* of state information whose primary copy is kept elsewhere, i.e., at coordinators. Replication through group communication is the main tool for enhancing availability of this information and for preserving its consistency in spite of failures.

Simulation results show that the protocol is indeed practical in that the latency cost induced by fault-tolerance in normal operating conditions, i.e., in the absence of failures, is limited to some milliseconds. Moreover the protocol exhibits very good scalability properties.

References

1. A. Acharya and B. R. Badrinath. A framework for delivering multicast messages in networks with mobile hosts. *ACM/Baltzer Journal of Mobile Networks and Applications*, 1(2):199–219, 1996.

2. S. Alagar, R. Rajagoplan, and S. Venkatesan. Tolerating mobile support station failures. In *Proc. of the First Conference on Fault Tolerant Systems*, pages 225–231, Madras, India, December 1995. Also available as Technical Report of the University of Texas at Dallas.
3. S. Alagar and S. Venkatesan. Causal ordering in distributed mobile systems. *IEEE Transactions on Computers*, 46(3):353–361, March 1997.
4. G. Anastasi and A. Bartoli. On the structuring of reliable multicast protocols for mobile wireless computing. Technical Report DII/00-1, Università di Pisa and Università di Trieste, January 2000. Submitted for publication. Available at http://www.iet.unipi.it/∼anastasi/papers/tr00-1.pdf.
5. G. Anastasi, A. Bartoli, and F.L. Luccio. Fault-tolerant support for reliable multicast in mobile wireless systems: Design and evaluation. Technical Report DII/02-1, Università di Pisa and Università di Trieste, March 2002. Available at http://www.iet.unipi.it/∼anastasi/papers/tr02-1.ps.
6. G. Anastasi, A. Bartoli, and F. Spadoni. A reliable multicast protocol for distributed mobile systems: Design and evaluation. *IEEE Transactions on Parallel and Distributed Systems*, 12(10):1009–1022, October 2001.
7. G. Anastasi and L. Lenzini. QoS provided by the IEEE 802.11 wireless LAN to advanced data applications: a simulation analysis. *ACM/Baltzer Journal on Wireless Networks*, 6(2):99–108, 2000.
8. V. Aravamudhan, K. Ratnam, and S. Rangajaran. An efficient multicast protocol for PCS networks. *ACM/Baltzer Journal of Mobile Networks and Applications*, 2(4):333–344, 1997.
9. A. Bartoli. Group-based multicast and dynamic membership in wireless networks with incomplete spatial coverage. *ACM/Baltzer Journal on Mobile Networks and Applications*, 3(2):175–188, 1998.
10. Ken Birman. The process group approach to reliable distributed computing. *Communications of the ACM*, 36(12):37–53, December 1993.
11. R. K. Budhia. Performance engineering of group communication protocols. Technical report, University of California, S. Barbara (USA), August 1997. Ph.D. dissertation.
12. V. Chikarmane, C. Williamson, R. Bunt, and W. Mackrell. Multicast support for mobile hosts using mobile IP: Design issues and proposed architecture. *ACM/Baltzer Journal of Mobile Networks and Applications*, 3(4):365–379, 1998.
13. F. Kaashoek and A. Tanenbaum. An evaluation of the Amoeba group communication system. In *Proc. 16-th IEEE-ICDCS*, pages 436–447, May 1996.
14. R. Prakash, M. Raynal, and M. Singhal. An efficient causal ordering algorithm for mobile computing environments. *Journal of Parallel and Distributed Computing*, March 1997.
15. G. Xylomenos and G. Polyzos. IP multicast for mobile hosts. *IEEE Personal Communications*, pages 54–58, January 1997.
16. L. Yen, T. Huang, and S. Hwang. A protocol for causally ordered message delivery in mobile computing systems. *ACM/Baltzer Journal of Mobile Networks and Applications*, 2(4):365–372, 1997.

JumpStart: A Just-in-Time Signaling Architecture for WDM Burst-Switched Networks*

Ilia Baldine[1], Harry G. Perros[2], George N. Rouskas[2], and Dan Stevenson[1]

[1] MCNC ANR, Research Triangle Park, NC, USA
[2] NCSU Department of Computer Science, Raleigh NC, USA

Abstract. We present an architecture for a core dWDM network which utilizes the concept of Optical Burst Switching (OBS) coupled with a Just-In-Time (JIT) signaling scheme. It is a reservation based architecture whose distinguishing characteristics are its relative simplicity, its amenability to hardware implementation, support for quality of service and multicast natively. Another important feature is data transparency - the network infrastructure is independent of the format of the data being transmitted on individual wavelengths. In this article we present a brief overview of the architecture and outline the most salient features.

1 Introduction

The adoption of dWDM as the primary means for transporting data across large distances in the near future is a foregone conclusion, as no other technology can offer such vast bandwidth capacities. The current dominant technology for core networks are wavelength-routed networks with permanent or semi-permanent circuits set up between end points for data transfer. Many of the proposed architectures treat dWDM as a collection of circuits/channels with properties similar to electronic packet-switched circuits with customary buffering (potentially done in the optical domain) and other features of electronic packet-switching. In addition, transport protocols used today (e.g. TCP) developed for noisy low-bandwidth electronic links, are poorly suited for the high-bandwidth, extremely low bit-error rate optical links. The round trip times for signaling and the resulting high end-node buffer requirements are a poor match to the all-optical networks characterized by the high bandwidth-delay product. In order to address the processing and buffering bottlenecks, characteristic of the electronic packet-switching architectures, and, by extension, their dWDM derivatives, a wholly new architecture is required, which is capable of taking advantage of the unique properties of the optical medium, rather than trying to fit it into existing electronic switching frameworks. In addition, in dWDM networks data transparency (i.e. independence of the network infrastructure from the data format, modulation scheme etc., thus allowing transmission of analog as well as digital signals) becomes not only possible, but desirable.

In this paper we present an overview of an architecture for a core dWDM network. The type of architecture, described in this paper, is wavelength-routed, burst-switching, with the "just-in-time" referring to the particular approach to signaling, taken within this

* This research effort is being supported through a contract with ARDA (Advanced Research and Development Activity, http://www.ic-arda.org).

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 1081–1086, 2002.

architecture. Signaling is done out of band, with signaling packets undergoing electro-optical conversion at every hop while data, on the other hand, travels transparently. For history of burst-switching the reader is referred to [6,9].

Just-In-Time (JIT) signaling approaches to burst-switching have been previously investigated in literature ([9,6]). The common thread in all these is the lack of the round-trip waiting time before the information is transmitted (the so-called TAG: tell-and-go scheme) when the cross-connects inside the optical switches are configured for the incoming burst as soon as the first signaling message announcing the burst is received. The variations on the signaling schemes mainly have to do with how soon before the burst arrival and how soon after its departure, the switching elements are made available to route other bursts. An example is the Just-Enough-Time (JET) scheme proposed in [7] which uses extra information to better predict the start and the end of the burst and thus use the switching elements needed to route the burst inside a switch for the shortest amount of time possible. Schemes also have been proposed for introducing QoS into the architecture ([8]). These schemes have been shown to reduce the blocking probability inside an OBS network with the disadvantage of requiring a progressively more complex schedulers ([5]).

In this short paper we present an overview and describe the salient features of the proposed Jumpstart architecture. For a more extensive treatment of the subject the reader is referred to [4,2].

1.1 Guiding Assumptions and Basic Architecture

The basic premise of this architecture is as follows - data, aggregated in bursts is transferred from one end point to the other by setting up the light path just ahead of the data arrival. This is achieved by sending a signaling message ahead of the data to set up the path. Upon the completion of data transfer the connection either times out or is torn down explicitly. Some of the basic architectural assumptions are summarized below:

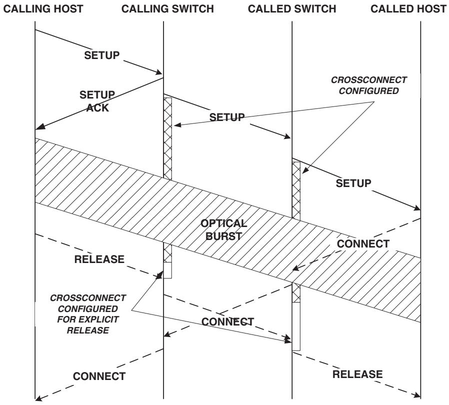


Fig. 1. Example of a burst

Out-of-band signaling - Signaling channel undergoes electro-optical conversions at each node to make signaling information available to intermediate switches.

Data transparency - *Data is transparent to the intermediate network entities,* i.e. no electro-optical conversion is done in the intermediate nodes and no assumptions are made about the data rate or signal modulation methods.

Network intelligence at the edge - Most “intelligent“ services are supported only by edge switches. Core switches are kept simple.

Signaling protocol implemented in hardware - So as not to create a processing bottleneck for high-bandwidth bursty sources, the signaling protocol must be implemented in hardware

No global time synchronization - In keeping with the “keeping it simple“ principle, we do not assume time synchronization between nodes.

A basic switch architecture presumes having a number of input and output ports, each carrying multiple wavelengths (envisioned to be in 100’s to 1000’s). At least one separate wavelength on each port is dedicated to carrying the signaling traffic. Any wavelength on an incoming port can be switched to either the same wavelength on any outgoing port (no wavelength conversion) or any wavelength on any outgoing port (partial or total wavelength conversion). The switching can be done by using MEMS micro-mirror arrays or some other suitable technology. Switching time is presumed to be in the μs range, with anticipation that it could be reduced further as the technology develops. Additionally, each switch is equipped with a scheduler which keeps track of wavelength switching configurations and configures the cross-connects on time to allow the data to pass through.

Data Transparency. We have briefly alluded to data transparency as being a desirable property of a core network of the future. Indeed, the ability to transmit optical digital signals of different formats and modulations, as well as analog signals simplifies many problems commonly associated with adaptation layers. In a burst-switched network, which essentially acts as a broker of time on a particular wavelength with high temporal resolution, this feature becomes relatively easy to implement, considering that signaling is done out of band on a separate channel. This is why JumpStart architecture makes no assumptions about the types of traffic it carries and instead schedules time periods on wavelengths within the network. The particular format that an end node uses to transmit its data to the destination is of no consequence to the network itself.

Processing Delay Prediction. Unlike data, signaling messages propagate through the network and accrue a processing delay inside each intermediate switch. For a SETUP message, which announces the arrival of a new burst to intermediate switches, this means that it has to be sent far enough in advance before the burst, so that the burst does not catch up with it before the destination is reached. Knowing this delay apriori, at the ingress switch, and communicating it to the source node (via SETUP_ACK) is part of the network function. This delay can be deduced from the destination address in the SETUP message, and further refined by the ingress switch over time, as CONNECT messages are sent back from the destination, indicating the actual processing delay incurred while the corresponding SETUP message traveled to the destination.

Quality of Service. When one talks about Quality of Service (QoS) in the context of contemporary packet-switching networking technologies and protocols (DiffServ, IntServ), the criteria for evaluating the QoS of a given connection involve network bandwidth and buffer management inside the routers and end-nodes. In the context of an all-optical transparent network such as proposed here most of these issues become irrelevant: data is transparent to the network and no buffering is done inside the network switches. The network acts merely as a time-broker on individual links. As a result when we discuss QoS in JumpStart, it is separated into several areas:

- QoS requirements defined for the specific adaptation layer used
- Optical QoS parameters, on which specific adaptation layer requirements may be mapped to
- Connection prioritization - allows us to preempt less important connections in favor of more important ones. It is a stand-alone property, which enables the network to deal with preemption of existing connections in a predictable manner.

Optical QoS parameters allow the network to route a connection along the best suited route depending on the type of signal the connection carries. Examples of optical QoS parameters are: bit-error rate, dynamic range, signal-to-noise ratio, optical channel spacing.

Multicast Support. Support for multicast connections is essential for future networks, however support for them within an architecture such as JumpStart may not be trivial. The optical signal must be split at certain points along the path according to the multicast routing tree in order for the network to remain all-optical, i.e. avoid electro-optical conversions. Such splitting presents a number of issues for the implementation, namely:

- A switch must be equipped with an optical splitting mechanism (splitting signaling messages does not present a problem, since they undergo electro-optical conversions in each switch).
- A number of such splits that can be done on a single connection is in general bounded by the optical power budget [3].

The result is that each connection may have a limited fan-out (contrary to present day electronic networks, where such issues are not considered).

Given these restrictions, for our network architecture we presume that the switches capable of splitting the optical signal are not common in the network, and, in fact, are sparsely dispersed throughout the network. Each end-node gets assigned one such switch as its multicast server switch through either administrative mechanism or a separate signaling mechanism. These special switches also take care of setting up routing trees for multicast connections, so in addition to special hardware they need to allow to split the optical signal, they also have special firmware to allow them to manage and route multicast connections. Thus all signaling messages from the source node that pertain to its multicast connections get routed by the network to its assigned multicast switch.

Within the multicast variety of connections we can identify two ways to setup a multicast session:

- Source-managed multicast: the source of the multicast knows the addresses of all of the members of the multicast group and that number is relatively small. In this case, the addresses of the members are directly included into the appropriate signaling messages by the source.
- Leaf-initiated join: in this scenario, a source may announce the existence of a multicast session, with a session id that is unique inside the network. Multicast servers in the network will learn of this new session through means that are outside of the scope of this discussion, and the end nodes will be able to join existing multicast sessions by communicating with their domain multicast servers.

In practice we would like to allow for a combination of both. A source may begin by specifying a few end nodes and allow the rest to use the leaf-initiated join capability. In the extreme case the source node simply announces an existence of a session and lets nodes in the network join as they wish. One additional option of multicast sessions is the session *scope*. The scope limits the availability of the session to nodes belonging only to specific domain(s). Additionally a source node may specify as part of the connection options that only it has the authority to add new leaves to the tree, in which case no leaf-initiated join connections will be allowed.

Label Switching. Label switching concept will be utilized by the signaling channel in order to achieve several goals:

- Speed up in accessing call state in the switch (based on label, not call reference number, which is not unique within the network).
- Guarantee that forward and backward routes coincide (while connections in a JumpStart OBS network are unidirectional, signaling paths are not, and it is necessary that signaling messages travel the on the same path both in the forward and in the backward direction).
- Speed up routing (once a connection path has been established, it is desirable that further signaling messages do not consult the routing table but use a pre-established path).

For this purpose, labels similar to MPLS will be used as part of the signaling message format. These labels will have link-local significance (unique on one link, but not within a switch or the network). Similar to ATM and MPLS, these labels will be rewritten as the signaling message traverses the network. Special tables within the switch will be needed to maintain the forward and the backward label mapping. No label stacking will be allowed.

Label distribution will be done on-the-fly, as part of signaling, while the connection is being setup, instead of utilizing a label distribution protocol like LDP or modified RSVP. Additional multicast support in a labeling mechanism will be necessary in those nodes that support multicast routing (multicast nodes). Unlike unicast-only nodes, which only need to maintain one-to-one label mappings for each connection, multicast nodes will require one-to-many and many-to-one mappings for label mechanism.

Persistent Connections. For some applications there will be a need for all bursts to travel the same route through the network, especially to those applications that are particularly sensitive to jitter or sequential arrival of information. Defining a persistent route service that precedes a series of bursts can allow the network to "nail down" a route for all subsequent bursts to follow. There are some network traffic engineering implications of persistent routes. If a significant portion of the network connections are established with fixed routes, then dynamic load balancing through routing changes in the network will become inefficient and perhaps fail. To minimize this potential problem, network service providers may choose to treat persistent route connections as a premium service offering. This service would be more expensive for service providers to support.

Multicast service as we have defined it also requires persistence. The first phase of establishing multicast service is to declare a session and build a routing tree. This is followed by one or several data transmission phase. Maintaining a persistent session is necessary so that the network can maintain state for multicast session routing as leafs are added and dropped through its lifetime.

1.2 Conclusions

In this paper we presented a short description of Jumpstart - a new proposed architecture for all-optical WDM burst-switched networks. We described and justified the need for the most important features in the network. Fore more information about the project we suggest [1].

References

1. Jumpstart project. In *http://jumpstart.anr.mcnc.org*.
2. Ilia Baldine, George Rouskas, Harry Perros, and Daniel Stevenson. Signaling Support for Multicasting and QoS within the Jumpstart WDM Burst Switching Architecture. *Optical Networks Magazine*, 2002. submitted for publication.
3. Karthik Chandrasekar, Dan Stevenson, and Paul Franzon. Optical Hardware Tradeoffs for All-Optical Multicast. In *Submitted to OFC*, 2002.
4. I.Baldine, G.N.Rouskas, H.Perros, and D.Stevenson. Jumpstart - a Just-In-Time Signaling Aarchitecture for WDM Burst-Switched Networks. *IEEE Communications*, page p.82, Feb. 2002.
5. Pronita Mehrotra, Ilia Baldine, Dan Stevenson, and Paul Franzon. Network Processor Design for use in Optical Burst Switched Networks. In *International ASIC/SOC Conference*, September 2001.
6. Chunming Qiao and Myungsik Yoo. Optical Burst Switching (OBS). *Journal of High Speed Networks*, 8, 1999.
7. Myungsik Yoo, Myongki Jeong, and Chunming Qiao. A High-Speed Protocol for Bursty Traffic in Optical Networks. In *SPIE*, volume 3230, pages pp.79–80.
8. Myungsik Yoo and Chunming Qiao. A New Optical Burst Switching Protocol for Supporting Quality of Service. In *SPIE*, volume 3531, pages pp.396–405.
9. John Y.Wei and Ray McFarland. Just-In-Time Signaling for WDM Optical Burst Switching Networks. *Journal of Lightwave Technology*, 18(12):pp.2019–2037, December 2000.

Device Discovery in Bluetooth Networks: A Scatternet Perspective

Stefano Basagni[1], Raffaele Bruno[2], and Chiara Petrioli[3]

[1] Northeastern University, Dept. of Electrical and Computer Engineering
basagni@ece.neu.edu
[2] CNUCE Institute, C.N.R. Pisa, Italy
bruno@guest.cnuce.cnr.it
[3] Università degli Studi di Roma, "La Sapienza," Dipart. di Scienze dell'Informazione
petrioli@dsi.uniroma1.it

Abstract. The paper concerns device discovery in multi-hop networks of Bluetooth devices. We start from the observation that forming a Bluetooth *scatternet* (i.e., a multi-hop wireless topology) requires each pair of neighboring nodes to have a "symmetric" knowledge of each other, i.e., if node u *knows* node v then node v knows node u. We investigate the use of the Bluetooth procedures for device discovery (*inquiry* procedures) in order to guarantee the needed symmetric knowledge for scatternet formation. Through the use of simulations we observed that despite the long time required for each node to become aware of the presence of all its neighbors, the Bluetooth topologies obtained by using the devices discovered after just 6 seconds are connected. The average number of neighbors of each node and the average route length are also consistently close to the values that we would obtain if all the neighbors of a device were discovered.

Keywords: Bluetooth networks, Device discovery, Scatternet formation.

1 Introduction

Bluetooth Technology (BT) [1] is emerging as one of the most promising enabling technologies for ad hoc networks.

When two BT devices come into each others communication range, in order to set up a communication link, one of them assumes the role of *master* of the communication and the other becomes its *slave*. This simple "one hop" network is called a *piconet*, and may include many slaves, no more than 7 of which can be active (i.e., actively communicating with the master) at the same time.

A BT device can timeshare among different piconets. In particular, a device can be the master of one piconet and a slave in other piconets, or it can be a slave in multiple piconets. Devices with multiple roles will act as gateways to adjacent piconets, resulting in a multihop ad hoc network called a *scatternet*.

Scatternet formation algorithms have been proposed in [2], [3], [4], [5]. These works have identified neighbor discovery (i.e., the process through which neighbors acquire a symmetric knowledge of each other) as the first and most time consuming operation to be performed by a BT device.

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 1087–1092, 2002.

A major problem is that the *inquiry procedures* provided in the BT specification for device discovery are time consuming and asymmetric. For two neighbor devices to handshake, they must be in "opposite" inquiry modes, namely one must be the inquirer, in *inquiry mode*, and the other device has to be willing to be discovered, i.e. it must be in *inquiry scan mode.* Also, the inquirer node is enabled to discover a neighboring device *without* having to identify itself to this device. In [4] and [5] "symmetric" methods for device discovery are proposed. In [5] each device alternates between inquiry and inquiry scan modes, randomly selecting the time to spend in each mode. In [4] time is divided into fixed length steps, and a node chooses randomly at each step whether to go in inquiry or in inquiry scan mode. When an inquirer node discovers one of its neighbors, a temporary piconet is created (by means of the *paging procedures*) so that the discovered neighbor can be made aware of the inquirer identity.

The Scatternet formation algorithms proposed in [4] and [5] rely on the assumption each nodes is in the transmission range of every other node ("single-hop" topology). This crucial assumption allows the device discovery phase to be fast and simple: There is no need for two neighboring devices to discover each other if this does not serve the purpose of the (centralized) scatternet formation protocols. The only solutions proposed so far which address the more general and practical case in which the original topology can be multi-hop, ([2] and [3]) require each node to become aware of its one-hop neighborhood.

In this paper we investigate the effectiveness of the device discovery scheme proposed in [5], and adopted in [2], for the most general case of multi-hop topologies. Simulation results show that the time for each node to be made aware of over 90% of its neighbors is over 18 seconds in case of dense networks. However, we also observed that after only 6 seconds the percentage of neighbors discovered is large enough to obtain connected topologies, i.e., connected scatternets. Knowing a smaller number of neighbors has also the desirable effect of lowering the number of slaves that a master has to manage. We finally present numerical results about the average length of routes (shortest paths) in the topology obtained by considering all nodes and the sole links corresponding to the discovered devices.

The paper is organized as follows. Section 2 describe the use of the inquiry and page procedures that allows at each node the symmetric knowledge of some of its neighbors. In Section 3 we describe the experimental results obtained by simulations and, finally, Section 4 concludes the paper.

2 Device Discovery in Bluetooth Networks

For a detailed description of the Bluetooth system, the reader is referred to [1]. In the following we focus on the inquiry procedures used for device discovery. A BT device that want to discover another BT device enters the *inquiry* substate. In this substate, it continuously transmits the *inquiry packet*[4] at different

[4] The inquiry packet is a packet that do not contain any information about the source, but only a general inquiry access code, GIAC.

hop frequencies. The inquiry hop sequence is always derived from the general inquiry access code. The inquiry response consists of the device in inquiry scan that transmits, after a backoff period necessary to avoid collisions with possible responses from other scanning devices, the *frequency hopping sequence*, FHS, packet with its own unique BT address and its BT clock. Notice that for each pair of neighboring devices u and v for which u discovered v the knowledge gained at each of the two nodes is "asymmetric." The node u (the inquirer) knows device v's access code (obtained from v's BT address) and BT clock. Device v knows nothing about device u.

The inquiry procedure described in the specification indicates how a device in inquiry mode can trigger a peer device in inquiry scan mode to send its ID and the synchronization information needed for link establishment. However, no indication is given on how to guarantee that neighboring devices are in opposite inquiry modes which is the needed condition for them to communicate these information to each other. Furthermore, the inquiry message broadcast by the source does not contain any information about the source itself, thus, once two neighboring devices complete an inquiry handshake, only the source knows the identity of the device in inquiry scan mode, not viceversa.

To overcome these drawbacks and attain mutual knowledge for each pair of nodes, we use a mechanism similar to that introduced in [5]. Each device is allowed to alternate between inquiry mode and inquiry scan mode. The time spent by each device in a given mode is uniformly distributed in a predefined time range (left unspecified in the BT specification). Hereafter, we describe the operations performed at each device during the topology discovery phase. The generic device v that executes the discovery procedure, sets a timer T_{disc}, which is decremented at each clock tick (namely, T_{disc} keeps track of the remaining time till the end of this phase). Device v then randomly enters either inquiry or inquiry scan mode, and computes the length of the next phase ($T_{\text{w inquiry}}$ or $T_{\text{w inquiry scan}}$). While in a given mode, device v performs the inquiry procedures as described by the BT specification. The procedures that implement the inquiry mode or the inquiry scan mode are executed for the computed time ($T_{\text{w inquiry}}$ and $T_{\text{w inquiry scan}}$, respectively), not to exceed T_{disc}. Upon completion of an inquiry (inquiry scan) phase, if $T_{\text{disc}} > 0$, a device switches to the inquiry scan (inquiry) mode. To allow each pair of neighboring devices to achieve a mutual knowledge of each others' ID and clock, our scheme requires that whenever a device in inquiry (inquiry scan) mode receives (sends) an FHS packet, a temporary piconet is set-up by means of a page phase. The master already knows ID and clock of the slave (through the inquiry phase). Setting up a piconet now ensure that the master send to the slave its FHS (i.e., its ID and clock) to the slave (this is accomplished through the slave and the master going into the slave response and master response substates, respectively). We notice that the temporary piconet set up time is extremely short, given that the two participating devices are already in the proper opposite paging modes (they do not have to find each other: the device in inquiry mode goes in paging mode right away, and the device in inquiry scan mode goes in paging scan mode immediately after inquiry response). Furthermore, the information to be exchanged is extremely short: The ID and clock of each device are included in the FHS packet, which is transmitted in one slot. As soon as this packet has been successfully transmitted the piconet is disrupted.

The effectiveness of the described mechanism in providing the needed mutual knowledge to pairs of neighboring devices relies on the idea that by alternating inquiry and inquiry scan mode, and randomly selecting the length of each inquiry (inquiry scan) phase (i.e., the values of $T_{\mathrm{w\ inquiry}}$ and $T_{\mathrm{w\ inquiry\ scan}}$), we have high probability that any pair of neighboring devices will be in opposite modes for a sufficiently long time, thus allowing the devices to discover each other.

3 Experimental Results

We have simulated the BT device discovery methods described in the previous section by using the VINT project Network Simulator ("ns2") [6] and BlueHoc [7], the IBM open-source extension to ns2 that implements the baseband and link layer of BT as described in the BT specification [1]. We have extended BlueHoc to provide: *i*) packet collision detection, *ii*) alternation between inquiry and inquiry scan, *iii*) determination if two nodes are neighbors based on their transmission radius and on their distance, and *iv*) dynamic selection of Master or Slave role at each node. We selected the $\mathrm{T_{w\ inquiry}}$ and $\mathrm{T_{w\ inquiry\ scan}}$ randomly and uniformly in the range $[t_{\mathrm{train}}, t_{in}]$ seconds, where t_{train} is the duration time of a single frequency train, and $t_{in} = 2$ (see also [5]). We have conducted experiments with $t_{in} = 4$ and $t_{in} = 6$ without observing significant variations with respect to the results reported below. All the simulations in the present section were run on a number of generated topologies large enough to achieve a confidence level of 95% with a precision within 5%.

3.1 Device Discovery in Multi-hop Networks

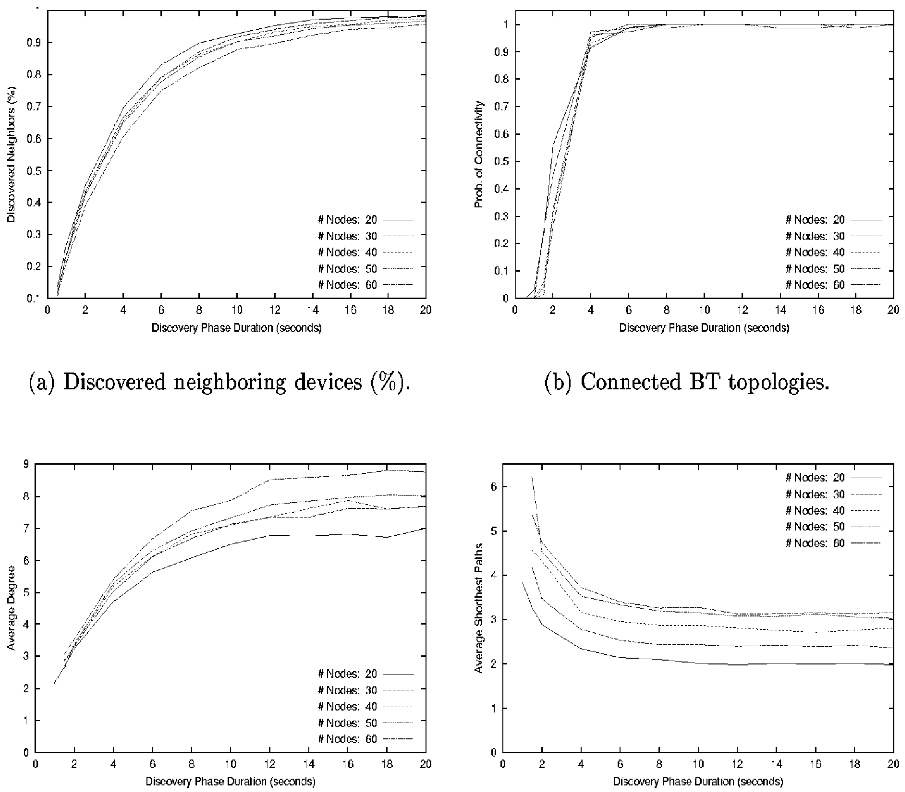
In what follows we term *original topology* the topology that we would obtain if each device could set up a bidirectional connection with all the devices in its transmission range (its neighbors). The term *BT topology*, instead, indicates the topology obtained by (bidirectionally) connecting only those neighbors that a device was able to discover in a predefined time T_{disc}.

Our set of experiments concerns the simulation of the device discovery procedure described above in networks of up to 60 BT devices. These networks are multi-hop in the precise sense that the radio vicinity of *all* devices is not required (as it is in the single-hop networks considered in [5] and [4]). The devices are scattered randomly and uniformly in a square area whose side L was chosen large enough to produce connected topologies with high probability. All experiments have been conducted on connected topologies. The properties of average degree and average shorthest paths are listed in Table 1.

Figure 1(a) shows the percentage of neighbors that each nodes locally discovers in at most 20 seconds in networks of 20 to 60 BT devices. The results are averaged over all the nodes in the network. We observe that the curves are very similar, given the similar average degree (i.e., the average number of neighbors

Table 1. Area dimension, average degree and average shortest path length

Number of BT devices	20	30	40	50	60
L	24	29	34	38	40
Avg. degree	6.982	7.851	8.058	8.378	9.213
Avg. shortest paths	1.882	2.264	2.666	2.966	3.065



(a) Discovered neighboring devices (%).

(b) Connected BT topologies.

Fig. 1. Some characteristics of the discovered BT topologies.

of each node), as listed in Table 1. We notice that it is not possible for a node, even in 20 seconds, to discover all its neighbors. However, Figure 1(b), shows that despite the number of device discovered is less than the number of possible neighbors in the original topology, when the original topology is connected, then the BT topology is connected as well, i.e., the possibility of obtaining a connected scatternet is not compromised. After 6 seconds the percentage of device discovered allows us to obtain a connected BT topology. Thus, the lower number of discovered devices could actually turn into “a blessing,” since connectivity is preserved and each node that will be a master has potentially less slaves to manage. The reduced degree is depicted in Figure 1(c). At around 6 seconds the average degree of the BT topologies is always less than 7, i.e., always less than the maximum number of active slaves that a master can handle. We observe also that the longer the time of the discovery phase, the closer the “BT degree” becomes to the original degree (Table 1). Finally, we computed the average shortest path length for both original topologies and their corresponding BT topology. The average shortest path length for the original topology is listed in Table 1. Figure 1(d) shows that after 6 seconds the duration of the discovery phase T_{disc} does not sensibly affect the average length of the shortest paths in the BT topology. As noticed for the BT degree, the average length of the “BT shortest paths” converge to the corresponding value for the original topology (Table 1).

4 Conclusions

In this paper we have considered the problem of neighbor discovery in multi-hop networks of Bluetooth devices. By means of extensive simulations we have shown that, despite the long time required for each node to become aware of the presence of all its neighbors, the Bluetooth topologies formed by devices discovered after just 6 seconds are connected, and do not result in significantly increased shortest paths between pairs of BT devices. Finally, we have shown that the length of the neighbor discovery phase is a powerful tuning knob to control the nodes degree, and therefore limit the number of slaves that a master has to manage.

References

1. http://www.bluetooth.com: Specification of the Bluetooth System, Volume 1, Core. Version 1.1 (2001).
2. Basagni, S., Petrioli, C.: Multihop scatternet formation for Bluetooth networks. To appear in: Prooceedings of the IEEE VTC Spring 2002, Birmingham, Alabama, May 6–9, 2002.
3. Zaruba, G., Basagni, S., Chlamtac, I.: Bluetrees—Scatternet formation to enable Bluetooth-based ad hoc networks. In: Prooceedings of the IEEE International Conference on Communications (ICC 2001). Volume 1., Helsinki, Finland, June 11–14 2001, pp. 273–277.
4. Law, C., Siu, K.Y.: A Bluetooth scatternet formation algorithm. In: Prooceedings of the IEEE Symposium on Ad Hoc Wireless Networks (SAWN 2001), San Antonio, Texas, 2001.
5. Salonidis, T., Bhagwat, P., Tassiulas, L., LaMarie, R.: Distributed topology construction of Bluetooth personal area networks. In: Proceedings of the IEEE INFOCOM 2001. Volume 3., Anchorage, Alaska, 22–26 April 2001, 1577–1586.
6. The VINT Project: The ns Manual. http://www.isi.edu/nsnam/ns/ (2001).
7. IBM: BlueHoc: Bluetooth Ad Hoc Network Simulator, Version 1.0. http://www-124.ibm.com/developerworks/projects/bluehoc (2001).

QoS Evaluation of Real-Time Applications over a Multi-domain DiffServ Experimental Test-Bed

G. Carrozzo, V. Chionsini, S. Giordano, and S. Niccolini

Department of Information Engineering University of Pisa
Via Diotisalvi 2 56126 Pisa Italy Tel. +39 050 568511, Fax +39 050 568522
{g.carrozzo,v.chionsini,s.giordano,s.niccolini}@iet.unipi.it

Abstract. This paper presents a QoS evaluation in a DiffServ experimental test-bed scenario. We implemented our field trial using prototypal routers running under Linux OS and we arranged them in order to make possible the interconnection with a remote island of a Multi-domain DiffServ network. The performance evaluation of Real-time applications presented in the paper will make clear how it is possible to provide "mission critical" applications with tool-quality level of service when appropriate algorithm and resource sharing are chosen and when these features are associated with a fair degree of aggregation. As a consequence the paper describes the results by means of a Mean Opinion Score (MOS) evaluation campaign to show how Real-Time applications (such as voice and video conferencing) may suffer for the lack of QoS.

Keywords: Experimental test-bed, Multi-Domain, DiffServ, Real-Time traffic.

1. Introduction

During the past years different proposals for Next Generation Internet architecture have been suggested. Integrated Services [1] and Differentiated Services [2] were the most promising ones. Unfortunately both of them showed their weakness when dealing with end-to-end QoS guarantees (in particular, IntServ lacks of scalability, and DiffServ lacks of "hard" guarantees). This research work intends to show how, by means of simple DiffServ mechanism applying to prototypal routers (edge and core DiffServ routers), it is possible to obtain satisfying results in terms of QoS parameters and in terms of user perceived quality. The IntServ access network is supposed to be unchanged compared with the framework of IntServ over DiffServ architecture [3]. The rest of the paper is organized as follows: in Section 1 we present our design and implementation of a real Multi-Domain DiffServ experimental test bed carried out in the framework of NEBULA project [4]. We show how DiffServ mechanism and our DiffServ aggregation strategies [5] well behave when dealing with Real-time application (mostly voice and video). In Section 3 the discussion is about the treatment of audio and video within the Real-time class. In Section 4 we analyze and comment the results highlighting the goodness of our aggregation strategies assumption. Finally we present our conclusion and future works.

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 1093-1098, 2002.

2. Test-Bed Description

This section presents our Multi-Domain test-bed, built-up in order to study the obtainable performance of a DiffServ Core Network when appropriate QoS mechanisms are used. We emulated three simple access domains interconnected by a DiffServ cloud by means of an ATM link in order to arrange the trial to be remotized. Our implementation is related to multiple site interconnection, in order to insert our trial in a more complex network (as in the scope of NEBULA project), where the study of QoS performance is more critical. The access domains are emulated by means of source and destination PCs connected to separate private networks. In Fig. 1 we detail our field trial; it is possible to distinguish the access domains where the sources and the destinations are located.

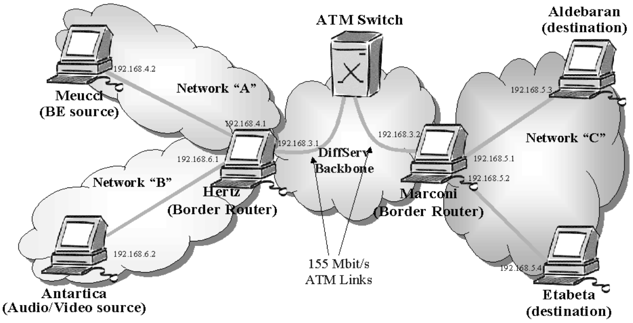


Fig. 1. Field Trial

We developed our field trial under Linux OS running on IA32 platforms (PC). The interconnecting routers are equipped with two 10/100 Ethernet cards and one MMF ATM card; each source/destination PC has only one 10/100 Ethernet card and it is point-to-point connected to the Border Router. The DiffServ backbone is emulated by means of an ATM connection (i.e. two 155 Mbit/s links towards a Newbidge CS1000 ATM switch). The Border Routers (Hertz and Marconi) provide the necessary transformation from packets to cells and viceversa by means of the AAL5 protocol. We implemented, on the BRs, the DiffServ traffic control functionalities (i.e marking, shaping, metering, dropping), by means of the “TC” package available under Linux. The scheduler used by the BRs is a CBQ (Class Based Queuing) algorithm.

3. Real-Time Traffic & Non Real-Time Traffic

The traffic used in this field trial may be classified in two main classes: a) Real time traffic: we include both audio and video sources because both of them have strict bounds on QoS target, even if different statistical features; b) Non Real-Time traffic: we include both MGEN (UDP source) traffic and FTP traffic (TCP source) because nor of them requires strict bounds on delay/jitter. The first test, whose results are presented in Section.4, provided a simple distinction between the two mentioned classes. The adopted combination between scheduling algorithm and TC functionalities had the aim to protect the Real-Time class against an "aggressive" and "persistent" BE class, formed by UDP sources. The second test, according to evaluations derived from a previous work [5], provided a refined classification, in order to avoid performance degradation experimented when voice and video flows are merged together.

4. QoS Evaluation

In all the tests presented in this section, Real-Time traffic was sent across the network. The first test conducted on the DiffServ backbone was about the flow isolation obtainable using the marking/scheduling algorithm on two classes (the first group of test is related to the transport of audio or video on the EF class while the rest of the traffic is forwarded on the BE class). We have collected the QoS relevant parameters directly between ingress and output interface of the ingress border router (named Hertz in Fig. 1), this because of synchronization problems arising from an end-to-end collection of delay.

Table 1. DiffServ EF configuration (video on EF)

Flow	Flow Description	ToS	CBQ Class Parameters
1	Video: 384kbit/s; Avg Packet size=800Bytes	EF	Buffer=3.2 kB;Rate=390 kbit/s
2	MGEN: rate=1Mbit /s; packet size=1kB	BE	Buffer=60kB;Rate=500 kbit/s

In the first test the scope was comparing QoS parameters of Real-Time Traffic, in terms of rate, delay and MOS in two cases: a) No DiffServ: all the traffic was sent on the same queue and served in a FIFO way; b) DiffServ: configuration described in Table 1. In Fig. 2 it is possible to notice that the portion of bandwidth used by the video flow increases when the protecting DiffServ scheduling scheme is activated. Fig. 3 shows that the performance degradation is more evident when speaking about the delay measurement. When the DiffServ is disabled all the flows share the same class and so the delay experimented by the video packets is the same experimented by the BE packets. On the other hand, when the video flow has its own queue (DiffServ enabled) we obtain the required service differentiation. In order to deeply analyze the performance of the video related to this test we have conducted a MOS (Mean Opinion Score) campaign; a MOS campaign is a collection of user sensation about quality perception by means of numerical score (1= lowest quality, 5= highest quality). This campaign has allowed to evaluate the perceived user quality at application level. We report in Fig. 6(a) the MOS evaluation obtained from the campaign when two video coding rates were considered: 128 kbit/s, 384 kbit/s. From the results it is possible to notice the application level performance improvement when DiffServ architecture is enabled.

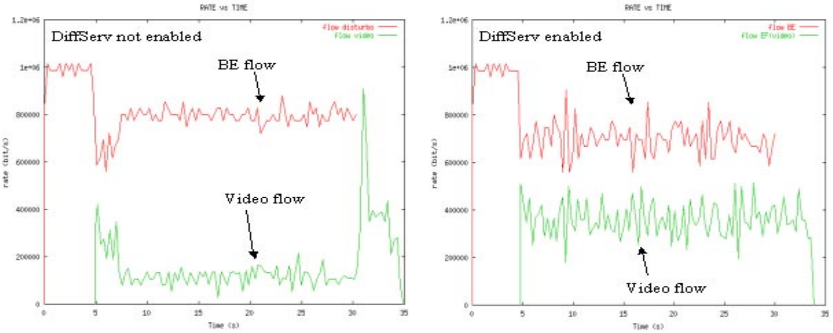


Fig. 2. Traffic rates: DiffServ disabled and enabled

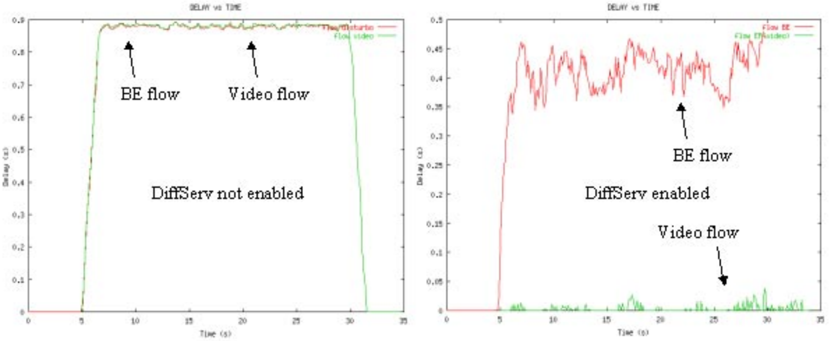


Fig. 3. Enabling the DiffServ mechanism on the video flow (delay)

A second test group is mainly focused on the evaluation of the impact of different aggregation strategies on the QoS parameters at network level and on the user perceived quality at application level. We will compare a scenario where only one class (EF) is used to carry Real-time traffic to a second scenario where we use separate classes in order to carry Real time flows (voice on EF and video on AF). The BE class is used in both cases to carry non-RT traffic. In Table 2 we show the scheduling parameters used in this test group.

Table 2. EF audio, AF video configuration

Flow	Flow Description	ToS	CBQ Class Parameters
1	Audio: 64kbit/s; Packet size=172 Bytes	EF	Buffer=1kB; Rate=64 kbit/s
2	Video: 128kbit/s; Avg Packet Size=800 Bytes	AF	Buffer=15 kB; Rate=128 kbit/s
3	MGEN: rate=1Mbit /s; packet size=1 kB	BE	Buffer =60kB; Rate =500kbit/s

The EF configuration adopted when audio and video are carried together may be obtained simply adding the CBQ parameters (i.e. Buffer = 16 kB and Rate = 192 kbit/s). We adopted this configuration in order to perform a fair comparison in terms of allocated resources. Fig. 4 shows the delay experimented by traffic flows when audio is carried on EF and video on AF, all collected on the DiffServ ingress BR. In this case the delay experimented by the audio flow is 35 μs (we had to zoom the statistic in order to make its visualization clearer) while the video delay is much more relevant (mean value: 25 ms). Video flow experimented a mean delay lower than BE flow one but higher than audio one, because of its different service class (AF) and its intrinsic burstiness. Anyway, the absolute values should not be considered, because they are relative to the crossing of a single device.

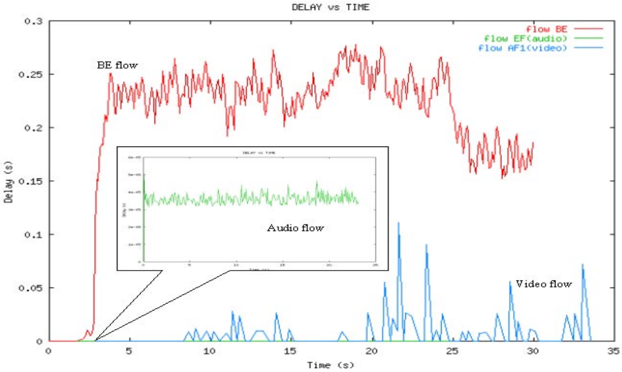


Fig. 4. Collection of delay parameter on BR (Audio on EF, Video on AF)

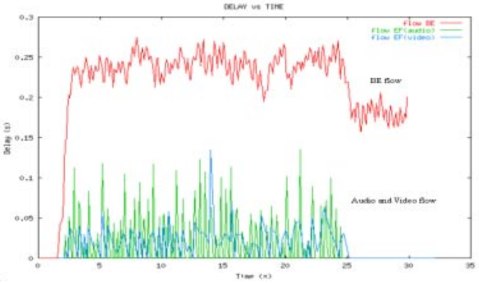


Fig. 5. Collection of delay parameter on BR (Audio and Video on EF)

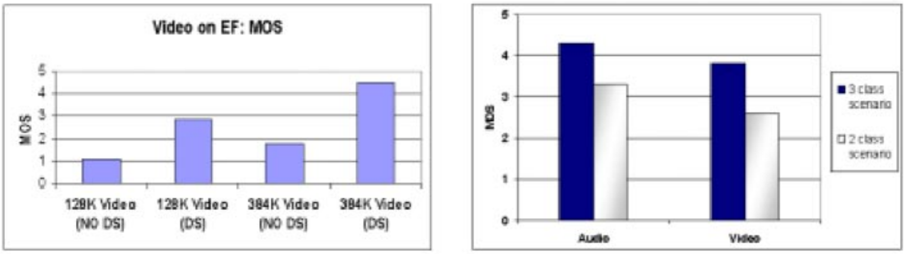


Fig. 6. MOS evaluation: a)Video on EF; b)Audio/Video with different aggregation strategies

When audio and video are merged together (EF class) there is a degradation of the audio performance: being carried together with the video, it suffers the same order of delay (see Fig. 5). At last we present in Fig. 6(b) the MOS evaluation collected in order to make a comparison between our proposed two-classes aggregation scheme and one-class aggregation scheme. Fig. 4 compared to Fig. 5 highlights the better performance obtained with the two-classes aggregation scheme. As it concerns the application level QoS, the MOS shown in Fig. 6(b) takes benefit from the forwarding of audio and video on two separate PHBs. As it can be noticed, the gap is approx. one point of MOS scale between the two scenarios: it is a great degradation of quality when speaking about user perceived quality at application level.

6. Conclusion and Ongoing Works

The main goal of the paper was the QoS evaluation of Real-time applications over a Multi-Domain DiffServ experimental test-bed by means of network level QoS parameters and application level parameters. In this framework the we have presented a two-classes aggregation scheme for DiffServ architecture in order to improve the obtainable performance. The collected results in the experimental test-bed scenario demonstrate they were satisfactory both at application level (evaluated by means of a MOS campaign) and at network level (evaluated by means of rate/delay statistics). This work is a first extract of our ongoing work on developing a real Multi-domain DiffServ island interconnection. A deeper analysis of end-to-end delays experimented in such a scenario is going to be conducted by means of a host synchronization tool (GPS system).

Acknowledgments. The authors wish to thank Andrea Giorgi for his help in this work. This work was partially supported by the project "NEBULA" of the Italian MURST.

References

1. R. Braden et al., "Integrated Services in the Internet architecture: An overview", RFC 1663, June 1994.
2. S. Blake et al., "An architecture for Differentiate Services", RFC 2475, December 1998.
3. Y. Bernet et al. "A Framework for Integrated Services Operation over Diffserv Networks", RFC 2998, November 2000.
4. NEBULA Project, financed by the Italian MURST (http://cofin98.cineca.it/murst-dae/).
5. R.G. Garroppo, S. Giordano, S. Niccolini, F. Russo, "A Simulation Analysis of Aggregation Strategies in a WF2Q+ Schedulers Network" in Proceedings of The 2nd IP Telephony Workshop, New York, April 2001.

A New Policy Based Management of Mobile IP Users

Hakima Chaouchi and Guy Pujolle

University of Paris VI, LIP6 networks Lab.
8 rue du capitaine Scott, 75015 Paris
{Hakima.CHAOUCHI ; Guy.pujolle}@lip6.fr

Abstract. A policy based management networking is a new paradigm used to achieve the network management. This paper presents a new policy based Mobile IP users management architecture based on a Common Open Policy Service (COPS) protocol which is currently deployed for QoS management. This paper introduces a new concept of terminal policy enforcement point (TPEP) which allows the terminal to interact with the network enforcing network policies defined by the network manager; it is a key feature of our architecture. The paper presents also the global architecture to support the mobile IP users requirements based mainly on two extensions of COPS protocol; COPS-SLS [1] for QoS negotiation and COPS-MU/MT [2] for policy based user and terminal mobility management.

1 Introduction

Due to the tremendous success of IP technology in the fixed network area, it is commonly accepted today that IP will provide the unifying glue for the increasingly heterogeneous, ubiquitous, and mobile environment [3, 4]. This paper presents new policy based architecture for user mobility management which supports nomadic users in the Internet by allowing them to access their personalized computing resources and services from anywhere on the Internet [5].

The IETF has proposed a policy based model for network management [6, 7, 8] and a TCP based policy transport protocol, called COPS (Common Open Policy Service) [9]. Policy based network management currently concerns QoS and security management. Many extensions have been introduced for COPS usage such as COPS-PR [9] for Diffserv, COPS-RSVP [10] for Inserv, and COPS-MIP [11] for Mobile IP terminal mobility management.

The next section presents an overview of a user mobility aspect and an overview of a policy based management networking. Then a new architecture to support Mobile IP users' management is presented followed by a conclusion.

1.1 User Mobility Overview

User mobility concerns *terminal mobility* and *personal mobility*. The *Terminal mobility* allows a terminal to change its network point of attachment without being disconnected from the network [12]. IP networks support terminal mobility using

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 1099-1104, 2002.

Mobile IP protocol. *Personal mobility* allows a user to use any available mobile or fixed terminal, and use his personal subscribed home network services from any terminal and any network access [13]. Thus, personal mobility is related to user location and service portability management [14]. A universal personal identifier is necessary to achieve personal mobility.

1.2 Policy Based Architecture Overview

Policy based management networking (PBMN) framework is proposed by the IETF [6, 7]. It is based on two important elements: policy server PDP (Policy Decision Point) and PEP (Policy Enforcement Point) as illustrated on Fig. 1. (a). PBMN intends to manage the network based on the business policies, these policies are translated to network policies and stored in the network. They are used to automatically configure the network elements to offer services based on the business level policies. The protocol used to exchange policy objects is COPS [9]. PDP and PEP exchange COPS messages which are detailed in [15] to achieve policy based network management. Fig. 1 (b) illustrates an example of PDP/PEP messages exchange process which are briefly explained below:
OPN: Client OPeN, PEP opens a TCP connection with the PDP; CAT: Client AccepT, the PDP accepts a connection; REQ: REQuest. PEP sends a request for a PDP. The request contains an identifier and policy objects necessary for a PDP policy decisions; DEC: DECision. The PDP sends a policy decision in a DEC message; RPT: RePorT. The PEP sends a report to the PDP after enforcing a policy contained in previous DEC message; CC: Client Close. The PEP closes a connexion.

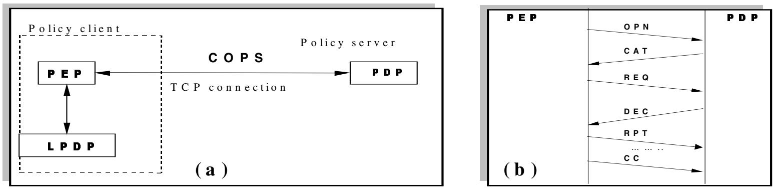


Fig. 1. (a) Policy based architecture. (b) PDP/PEP COPS messages exchange.

2 A New Policy Based Mobile IP Users' Management Architecture

We identify four issues related to user mobility management which are user registration, terminal registration, service portability and QoS negotiation.
To achieve these challenging issues, we introduce new components in the IETF policy based architecture illustrated on Fig. 2 and we introduce COPS extension called COPS-MU/MT [2] (COPS-Mobile User/Mobile Terminal) which defines new policy objects to support user and terminal registration respectively, user service portability, and QoS negotiation.

2.1 Architecture Components

Some Mobile IP terms [16] are necessary to understand the next sections, they are explained bellow:

HA. Home Agent, maintains a mobility binding of a MT in his home network.

FA. Foreign Agent maintains a list of terminal visitors in the foreign network.

Mobility binding. It's an association between the Home address and the CoA of a mobile terminal;

Home address. It's a routable and a permanent address used to locate a mobile terminal even when it changes its point of attachment. It is a HA adress.

CoA. Care of Address, is the address obtained in a foreign network. CoA may be a FA address (IPv4) or a co-located CoA (IPv6) [16]. If a mobile terminal has a co-located CoA, it interacts directly with the HA else, it interacts with the FA which forwards its messages to the HA.

Fig. 2 illustrates new components introduced in COPS-MU/MT architecture.

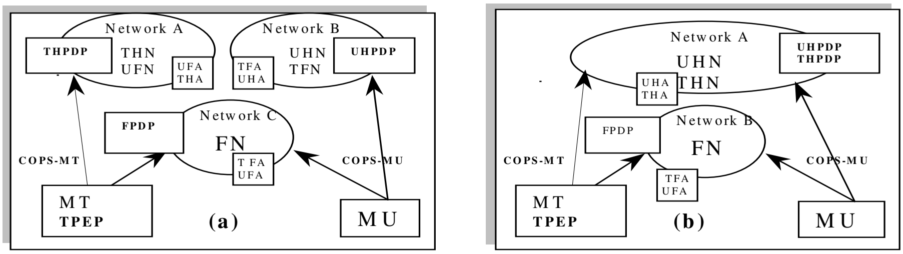


Fig. 2. Terminal's and user's home and foreign networks. **(a)** MU and MT are subscribed in different networks. **(b)** MU and MT are subscribed in the same network.

Some components defined for a Mobile User (MU) and a Mobile Terminal (MT) have similar functions such as a:

A User Home Policy Decision Point (**UHPDP**) and a Terminal Home Decision Point (**THPDP**) which are policy servers in a User Home Network (**UHN**) and a Terminal Home Network (**THN**) respectively.

A User Foreign Policy Decision Point (**UFPDP**) and a Terminal Foreign Policy Decision Point which are policy servers of a User Foreign Network (**UFN**) and a Terminal Foreign Network (**TFN**) respectively.

A Foreign policy Decision Point (**FPDP**) is a policy server of a Foreign Network (**FN**) of a mobile user and a mobile terminal.

A key feature of our architecture is a Terminal Policy Enforcement Point (**TPEP**). It is introduced to allow the terminal to interact directly with the network for user and terminal registration, QoS negotiation and user service portability.

User Home Agent (**UHA**) and Terminal Home Agent (**THA**) maintains the user and the terminal mobility binding respectively.

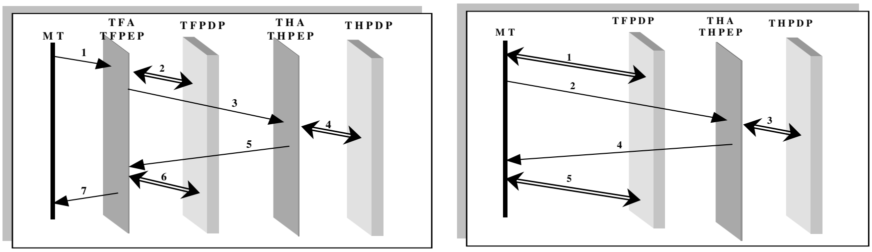
User Foreign Agent (**UFA**) and Terminal Foreign Agent (TFA) are the FA of the mobile user and the mobile terminal respectively.

Policy servers of different network providers have to maintain policy information related to home and foreign mobile users such as user profile, services profile, and terminal profile in order to allow a user universal access to network services and resources from anywhere. The different profiles may be stored in the home agents or in the policy servers.

The goal of the policy based Mobile IP user management is to allow the user to access his home services with the parameters negotiated with his home network from anywhere. Thus, the policy based Mobile IP users management achieves the user and terminal registration to support user and terminal location management and the personal service portability and the QoS negotiation to support user services anywhere.

2.2 Policy Based User and Terminal Registration

Terminal registration. Terminal registration must be achieved only if a terminal is located in a foreign network, if it is a fixed terminal or a mobile terminal located in its home network then a terminal registration is unnecessary. **COPS-MT** is used to achieve the terminal registration, it supports IPv4 and IPv6 registration by allowing the **TPEP** directly interact with the **FPDP** so that the mobile terminal can achieve its registration directly with the HA. Fig. 5 illustrates COPS-MT terminal registration related to FA CoA (IPv4) and co-located CoA (IPv6).



(a) **(b)**

Fig. 3. COPS-MT terminal registration. (a) FA CoA. (b) Co-located CoA.

Numbered steps illustrated on Fig. 3 (a) correspond to the terminal registration in Mobile IPv4 with FA CoA whereas Fig. 3 (b) corresponds to the case when a mobile terminal has a co-located CoA such as in IPv6. Fig. 3 (b) steps are explained bellow:

1. TPEP interacts directly with TFPDP for terminal registration request policy decisions;
2. MT sends registration request to the THA;
3. THPEP interacts with THPDP for terminal registration request;
4. THA sends registration reply message to the MT;
5. TPEP interacts with FPDP for terminal registration reply policy decisions.

The steps described on Fig.3 (a) are related to the case where a MT has a FA CoA, they are different from steps in Fig.3 (b) in that a FA intercepts messages sent by a MT and forwards them.

User registration. A user registration must be achieved every time a user logs in a terminal even if a user is in his home network. In **COPS-MU**, the user mobility registration is similar to COPS-MT terminal mobility registration. COPS-MU user registration consists of maintaining an association between the terminal home adress and a user identifier. The necessary elements for achieving user registration are UHPDP, UFPDP, UHA and UFA. The registered user would be reachable on the terminal he is using and may use his home services from anywhere.

2.3 Policy Based Mobile IP User Service Portability and QoS Negotiation

In this work we assume that the network is a policy based network QoS management such as Diffserv COPS-PR policy provisioning based network and we propose to support a QoS negotiation for a Mobile IP user which moves from the home network to a foreign network. In this architecture **COPS-MU/MT** is deployed in a wireless access network to achieve a user and terminal registration and QoS negotiation, and COPS-SLS [1] is deployed between the home PDP and a FPDP for inter-domain negotiation of a user home subscribed QoS and COPS-MU for inter-domain mobile user and mobile terminal registration and mobile user service portability negotiation.

When a mobile user moves to a foreign network, a FPDP interacts with the UHPDP to determine the user's QoS negotiated with the home network so that the mobile user has not to re-negotiate a QoS with the foreign network. This architecture is illustrated in Fig. 4.

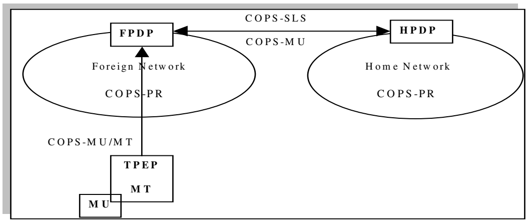


Fig. 4. Policy based Mobile IP users QoS negotiation environment.

This architecture is also used to negotiate user service portability. The FPDP negotiate with the UHPDP where to run the user personal services. The UHPDP decides based on the user profile, the personal service profile and the terminal profile where to run the user personal service. This part will not be detailed in this paper.

3 Conclusion

In this paper, we have described new policy based architecture to support user mobility management in IP networks. The approach taken assumes that mobile users are in IP networks based on the PDP/PEP architecture and using COPS protocol. We have proposed to use a terminal Policy Enforcement Point (**TPEP**) which allows the terminal to interact directly with the appropriate PDP and we proposed also the COPS extension named **COPS-MU/MT** (COPS-Mobile User/Mobile Terminal) to support a policy based user mobility management issues related to user and terminal registration, user services portability and QoS negotiation.

We believe that the use of COPS and PDP/ PEP model offers a good way to achieve a unified IP network policy management of QoS, security, mobility, etc. However, we need to implement this architecture for performance evaluation.

Future work intends to define all necessary policy objects related to user mobility registration, terminal mobility registration, service portability, and QoS negotiation.

References

[1] M. Nguyen, "COPS-SLS", IETF-draft, november 2001, draft-nguyen-rap-cops-sls-01.txt

[2] H.CHAOUCHI, G. Pujolle, « COPS-MU : a new policy based user mobility management », proceeding MS3G 2001, Lyon, France.

[3] A. Fasbender,F. Reichert, E. Gueulen, J. Hjelm, T. Wierelemann, "Any Network, Any Terminal, Anywhere", IEEE Personal Communications 1999.

[4] L. Bos, S. Leroy, "Toward an ALL-IP-Based UMTS System Architecture", IEEE Network, January/February 2001.

[5] Yalun Li, V. Leung, "Protocol Archirtecture for universal personal computing", IEEE Journal on selected areas in communications, vol 15, N° 8, October 1997.

[6] A. Wtersinen, J. Schnizelein, J;Strassner, M. Scherling, B. Quinn, J. Perry, S. Herzog, A. Huynh, M. Carlson, S. Waldbusser, "Policy Terminology", Internet Draft, March 2001, draft-ietf-policy-terminology-02.txt

[7] B. MOORE, E. Ellesson, J. Strassner, A. Westerinen, "Policy Core Information Model", RFC 3060, February 2001.

[8] M. Fine, K. McCloghrie, J. Seligson, K. Chan, S; Hahn, R. Sahita, A. Smith, F. Reichmeyer, " Framework Policy Information Base", Internet draft, March 2001, draft-ietf-rap-frameworkpib-04.txt.

[9] K. Chan, J. Seligson, D. Durhan, K. Mcloghrie, S. Herzog, F. Reichmeyer, R. Yavatkar, A. Smith "COPS usage for Policy provisioning (COPS-PR)", RFC 3084, March 2001.

[10] S. Herzog, J Boyle, R.Cohen, D.Durham, R.Rajan, A.Sastry, "COPS usage for RSVP", RFC 2749, January 2000.

[11] M. Jaseemuddin, A. Lakas, 'COPS usage for Mobile IP ', Internet draft, October 2000, draft-ietf-jaseem-rap-cops-mip-00.txt.

[12] G. Forman, J. Zahorjan, " the challenge of mobile computing", IEEE Computer, March 1994.

[13] E. Koukoutsis, C. Kossidas, N. Polydorou, " User Aspects for Mobility", Acts Guideline SII-G8/0798.

[14] M. Cristina Ciancetta, G. Colombo, R. Lavagnolo, D. Grillo, F. Bordoni, "Convergence Trends for fixed and mobile services", IEEE Personnal Communications, April 1999.

[15] D. Durham, J. Boyle, R. Cohen, S. Herzog, R. Rajan, A. Sastry, "The COPS (Common Open Policy Service) Protocol", RFC 2748, January 2000.

[16] C. Perkins, "IP mobility support", RFC 2002, October 1996.

A Framework for Policy-Based Management of QoS Aware IP Networks

P. Cremonese[1], M. Esposito[2], S. Giordano[3] M. Mondini[3], S.P. Romano[2], and G. Ventre[2]

[1]NETIKOS - via Matteucci, 56100 Pisa
piergiorgio.cremonese@netikos.com

[2]DIS -- Dipartimento di Informatica e Sistemistica, Università di Napoli Federico II
Via Claudio 21, 80125, Napoli, ITALY
{mesposit, spromano, giorgio}@unina.it

[3]ICA -DSC- EPFL - CH-1015 Lausanne, Switzerland
silvia.giordano@epfl.ch,

Abstract. In today's Internet, Policy-Based Network Management is gaining more and more proselytes. Its appeal is due to the given opportunity of a standard and consistent way for network configuration, independently of the underlying architecture and Quality of Service (QoS) provisioning model assumptions. The event-driven paradigm, well established in the general-purpose programmers world, through the Policy-Based approach begins to play its role also in the field of network management. In this paper we describe a policy framework suited for dynamic network management in QoS-enabled IP networks. First, we design an object model conceived to represent policies in a network-independent fashion. Then, we describe a management and configuration system based on Common Open Policy Service (COPS). Finally, we show a system prototype pointing out the main features of a Differentiated Services network management and configuration based on Policy System Management.

1 Introduction

A policy is a set of rules or methods, representing an object behavior or a decision strategy to be applied in order to ultimate a particular goal. The Policy-Based Network Management is the application of these organizational policies in order to manage the networks. With this approach, the role of network management moves from passive network monitoring to active QoS (Quality of Service) and network service-level-agreement provisioning .

While this technology is powerful and alluring, it's also generally untested and unproven. Worse, this area still suffers from a lack of standards and for a lack of ad hoc use of existing ones. There are two key issues that are not yet totally addressed: first, how the vendors will access and control their hardware, and second, how these systems glean information about an organization's users and resources.

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 1105-1110, 2002.

Our architecture, developed in the framework of the European IST project CADENUS, tries to address all those problems. First, we are developing a prototype that aims to test and validate the policy-based approach in a real DiffServ network. Secondly, we adopted a layered model. In this way, at the lower layer, we accomplished the devices configuration by employing a combination of CLI (command-level interface), COPS and LDAP. We feel that our work can be a step toward a standardized policy-based network management.

This document is specifically concerned with the definition of the processes that take place right after a new Service Level Specification (SLS) has been created as a consequence of the negotiation of a new service instance between, for example, an end-user and a Service Provider (SP). We are not focussing on the interactions that bring to the creation of an SLS, but simply assume that a new SLS has been provided by a Service Provider stemming from an even higher service level description (see [SLA]).

This document is organized in five sections. Next section illustrates the proposed architecture and the steps performed from an SLS to the final configuration of the devices. The Multiple-Layer, Policy-Based approach, with particular attention to the policies repositories used at each layer, is presented in Section 3. Section 4 expands on the Network Controller, which represents one of the main components of the overall architecture. Finally, Section 5 provides some concluding remarks, together with a discussion of future work.

2 A Framework for Automatic Configuration and Management of QoS-Aware Networks

Policy-Based Management has been thought to allow network configuration in the sphere of several applications, ranging from security and network engineering to monitoring and measurements. In this work, we will delve into the role of policies with respect to Quality of Service (QoS) needs in a QoS-aware network. In order to make network configuration and management an automatic task, independent of the specific devices implementing the network, our architecture is composed of three layers, each related to a different level of abstraction. More precisely, as depicted in Figure 1, the overall process starts from an abstract service description (contained inside an SLS), and comprises a number of intermediate steps, each needed in order to lower the level of abstraction, thus filling the gap between the human-oriented concept of a “service” and the device-specific configuration commands that eventually enforce the service itself. For each domain (i.e. Autonomous System --- AS) we have one functional block, named Resource Mediator (RM), that is in charge of managing the whole underlying network.

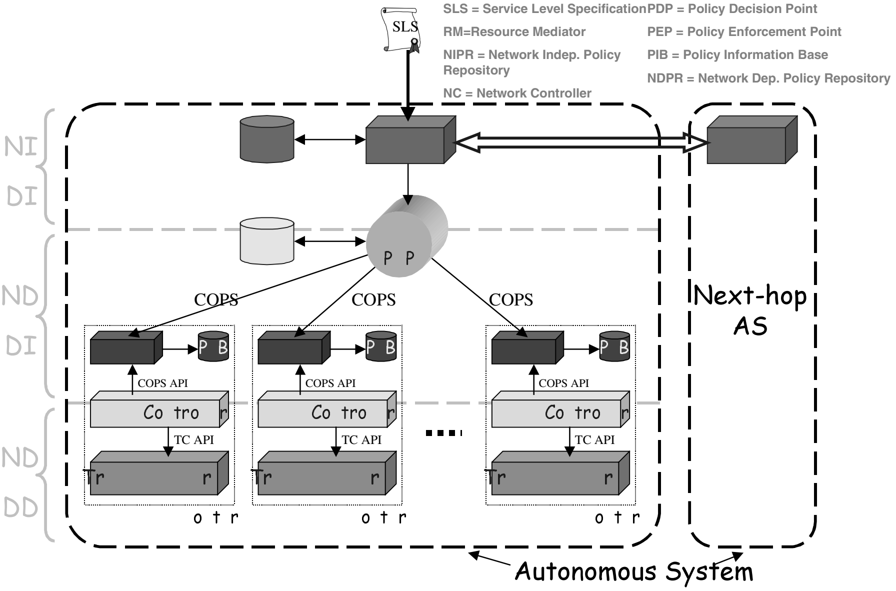


Fig. 1. Different layers of the Policy Architecture

The scenario we analyze is one in which, starting from an SLS instance, we go all the way down through the shown components in order to arrive at the network devices and appropriately configure them. Delving into the details of such a process, we identify the following steps:

1. A Resource Mediator takes an SLS and translates it into a coherent set of Network Independent Policy Rules (NIPR). As the name itself suggests, such rules are to be both network and device independent: they just are a well-structured representation of the information contained inside the SLS. The model we are thinking to adopt is inspired to the various proposals stemming from the Common Information Model [CIM] under standardization inside both the IETF and DMTF research communities [PCIM],[PCIMe],[PQIM].

2. The Network Independent Policy Rules are then passed to a Network Controller (NC), which translates them on the basis of the specific network architecture adopted (MPLS, Diffserv, etc.). The translation process brings to a new set of rules, named Network Dependent Policy Rules (NDPR), that are stored inside an ad-hoc defined Policy Information Base (PIB). The NC also acts as a Policy Decision Point (PDP) [COPS], which exploits a protocol like COPS to send, based on the "provisioning" paradigm, policies to the underlying Policy Enforcement Points (PEPs).

3. Upon reception of a new policy, the PEP is now in charge of interacting with a Device Controller (DC), thus triggering the last level of translation, so to produce the necessary configuration commands needed to appropriately configure the traffic control modules (e.g allocation and configuration of queues, conditioners, markers, filters, etc.) on the underlying network elements.

3 The Policy-Based and Multiple-Layers Approach

As we introduced in the previous sections, the innovative aspect of the CADENUS architecture is the policy-based and three-layers approach. The sequential steps performed by each layer aim to achieve the ultimate goal of setting up the network in an automatic fashion, without any human intervention. The SLS is an abstract service description, independent both of the network architecture (e.g. Diffserv, MPLS, ATM) and of the devices architecture (e.g. a CISCO router, a PC running Linux or FreeBSD). Yet, network-dependent and device-dependent information is still needed in order to configure and manage the network devices. For this purpose our architecture includes three databases containing the views of the requested QoS at different layers: the Network Independent Policy Repository, the Network Dependent Policy Repository, and the Vendor Dependent Policy Repository.

3.1 NIPR: A Repository for Network Independent Policies

The Network Independent Policy Repository (NIPR) is an archive located at the Resource Mediator (RM) level. When a new SLS arrives at the RM from the Service Provider, the RM translates it in a set of policy rules describing conditions and actions related to the requested service (still in a network-independent form) and stores it in the NIPR. This SLS must describe the single service instances in an unambiguous fashion.

The peculiarity of such an approach is that a NIPR, being at a high level of abstraction and then entirely network independent, can represent a common component (for every network architecture) containing the bundle of services to be enforced in the future. Anyway, as already stated, the semantic value of information stored in the NIPR, is not different than the one contained in the original SLS: the only added feature is the policy-based representation.

3.2 NDPR – A Repository for Network Dependent Policies

As we just explained, the Network Independent Policy Repository is a formal representation of the information contained inside an SLS. It contains, in standard format, the otherwise fuzzy definition of a *service*. In order to let such a definition become comprehensible to the lower network management devices, the need arises for a further level of translation. For this step, the Network Controller (see figure 1) goes a step further, by taking into consideration the specific network architecture that will support the deployment of the service. These network dependent policies are stored in the Network Dependent Policy Repository (NDPR). The NDPR contains policies in a representation independent of the devices implementing the network. It introduces, in the service/flow description, rules deriving from the supported technology (e.g. Diffserv) without going in detail of devices characteristics and components. NDPR generation is performed by Network Controllers according to business rules defined for traffic classification. The mapping from NIPR (which is based on user/service requirements) to NDPR (based on network implementation) is local to each domain. Each NC uses some business rules for policy generation and policy distribution. Such a policy could, for example, lead to the marking (via DSCP field) of a packet, or dropping, remarking, delaying of out of profile packets.

A policy is defined by the instantiation of a filter object (condition) and an action object. A filter object identifies, for instance, source and destination, while an action object can include Classifier object, Meter object, Shaper object.

3.3 VDPR – A Repository for Vendor Dependent Policies

This layer works with a representation that can be understood and handled by devices, thus reflecting their specific characteristics. The policies defined at the previous layer are translated into device configuration policies. Information, which is vendor and device dependent, such as queues configuration and network interfaces, is added at this layer. The vendor dependency derives from the necessity to make rules according to the specific features of the managed device. The schema for the translation of PIBs changes with the device nature. Therefore, this translation is demanded to a dedicated component, named the *Device Controller*. In our case, this component has been implemented for Linux-based routers, exploiting the functionality made available by the Linux Traffic Control (TC) module.

4 The Network Controller

The Network Controller (NC) is the component responsible for network management and configuration. Each NC manages a homogeneous network, where "homogeneous" means that only one technology for QoS support is provided within the network. The NC role can be summarized as follows:

- it performs management and configuration based on requests coming from the RM;
- it provides the RM with data for updating local repositories (routing, resources);
- it provides input to devices for local Traffic Control configuration;
- it manages the network with respect to fault detection and SLA monitoring.

The main tasks the NC has to accomplish refer to policy generation and instantiation.

4.1 Policy Generation

The NC receives a request for subscription from the RM, related to a service to be committed. The request is composed of a set of policies Network Independent (NI). The NC translates all involved policies in a network dependent format; it checks the consistency of these policies (e.g. availability of the requested resources) and sends the answer back to the RM. The generated set of policies will be stored in the NDPR.

4.2. Policy Instantiation

In this phase, the NC identifies the involved devices and sends (via COPS) the set of policies related to the request to the corresponding Device Controllers (DC). The DCs will in turn translate the received policies into the right traffic control commands needed to appropriately configure the network devices.

5 Conclusions and Future Work

In this paper we have shown an innovative approach for QoS-aware network configuration by means of policies. Such an approach has the advantage to provide a completely general way to achieve end-to-end QoS guarantees. Thanks to its layered structure, the architecture we propose is capable to make an adaptation from a service instance representation, as it is perceived at an abstract level, to a set of commands to be enforced on the underlying QoS-aware network nodes.

This architecture is going to be implemented as prototype and tested in the framework of the European project CADENUS. The main goal of this work will be:

- to emphasize the power and attractiveness of the proposed technology;
- to show its validity by means of a prototype;
- to give results of tests and trials;
- to identify current lacks and propose solutions;
- to accelerate the step toward a standardization of all the elements of policy-based management network.

References

[CIM] Distributed Management Task Force, Inc., "*Common Information Model (CIM) schema*", version 2.3, March 2000.

[PCIM] J. Strassner, E. Ellesson, B. Moore and A. Westerinen, "*Policy Core Information Model - Version 1 Specification*", RFC3060, February 2001.

[PCIMe] B. Moore, L. Rafalow, Y. Ramberg, Y. Snir, J. Strassner, A. Westerinen, R. Chadha, M. Brunner, R. Cohen, *"Policy Core Information Model Extensions"*, <draft-ietf-policy-pcim-ext-01.txt>.

[PQIM] Y. Snir, Y. Ramberg, J. Strassner, R. Cohen, "*Policy Framework QoS Information Model*", Internet draft, <draft-ietf-policy-qos-info-model-01.txt>, April 2000.

[COPS] J. Boyle, R. Cohen, D. Durham, S. Herzog, R. Rajan and A. Sastry, *"The COPS (Common Open Policy Service) Protocol"*, RFC2748, January 2000.

[SLA] S.P. Romano, M. Esposito, G. Ventre and G. Cortese, *"Service Level Agreements for Premium IP Networks"*, work in progress, Internet Draft <draft-cadenus-sla-00.txt>, available at http://www.cadenus.org/papers, nov 2000.

[SLS] D. Goderis, Y. T'joens, C. Jacquenet, G. Memenios, G. Pavlou, R. Egan, D. Griffin, P. Georgatsos, L. Georgiadis, P. Van Heuven, "*Service Level Specification Semantics, Parameters and negotiation requirements*", Internet-Draft, <draft-tequila-sls-01.txt>, work in progress, June 2001, expires December 2001.

SIP-H323: A Solution for Interworking Saving Existing Architecture

G. De Marco[1], S. Loreto[2], G. Sorrentino[3], and L. Veltri[3]

[1]University of Salerno - DIIIE- Via Ponte Don Melillo - 56126 Fisciano(Sa) – Italy
Ph.: + 39 0974 824700, Fax: +39 0974 824700, gdemarco@unisa.it
[2]Ericsson Lab Italy – Via Madonna di Fatima, 2 - 84016 Pagani – Italy
Ph.: + 39 081 5147733, Fax: +39 081 5147660,
salvatore.loreto@eri.ericsson.se
[3]Coritel - Via Anagnina, 203 - 00040 Roma – Italy
Ph.: + 39 06 72589169, Fax: +39 06 72583002,
{sorrentino,veltri}@coritel.it

Abstract. In the 3rd generation multimedia communication world and in the 3GPP standardization consortium, SIP protocol appears to be the preferred signaling protocol. However, the need to communicate with non-SIP based network, e.g. H.323 from ITU-U, is still a reality. The need can be satisfied with the introduction of network gateways (also named Inter-Working Function). One of the open issues about SIP-H.323 interworking is the address resolution, in other words, the automatic forwarding of a SIP call to an H.323 user. The paper proposes a SIP network architecture which can interoperate with H.323 networks, safeguarding the existing software/hardware components, as SIP terminal clients or SIP server proxies or IWF gateways.

Keywords: Sip, H323, interworking, gateway, call routing

1 Introduction

One of the problems arising in the future multimedia network is the interworking between networks that use different protocols; at present days, such problems mainly concern the interworking between SIP (developed in IETF) and H.323 (from ITU) based multimedia networks. Both protocol are signaling protocol, being currently H323 the standard for any IP based implementation of multimedia communications ([1],[2],[3]). The 3GPP has selected SIP as the signaling protocol for multimedia communications in the UMTS network. All these considerations lead to the conclusion that the interoperation of H.323 and SIP based networks is becoming a very crucial problem ([5], [6], [7]). Among the various problems that arise when considering the interworking of these two protocols, one important aspect is to allow, for example, a SIP user to reach a remote user on both SIP and H.323 networks; of course if the remote terminal is an H.323 terminal, then an interworking system (gateway) is needed. A satisfactory solution, involving additional protocols [4]. In this work, we propose a new interworking solution that requires no modifications of these network elements. The proposed solution is so based on the assumption that neither the client applications (the terminals) nor the network servers/gateways should be

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 1111-1116, 2002.

modified. Let us consider for example the following scenario: the owner of a big SIP network (> 500 consumers) has already acquired all the necessary servers; he/she has already installed and configured all the multimedia terminals; moreover, he/she has acquired the network nodes/servers and the H.323/SIP gateways (in the following referred also as *interface module* or *Inter Working Function*). Modifying the gateway source code or asking for a new version should be too expensive. In this context, we will see how to solve the addressing and registration aspects of the interworking problem without implementing the TRIP protocol within the SIP and H.323 signaling servers.

The main idea consists of the introduction of a new network component that easily allows the call forwarding from SIP to SIP domains or from SIP to H.323 domains.

2 System Outline

In a pure SIP network, terminals are named UserAgents (UA), while the servers can be classified as SIP Proxy servers, Redirect servers, and Registrar servers [8]. In our scenario, we consider a SIP network composed of UAs, stateful SIP Proxies acting also as Registrar servers, and one or more gateways (GWs) to other non-SIP IP networks. In such a network scenario, SIP terminals communicate directly with other SIP terminals and via an appropriate GW with non-SIP terminals. SIP Proxy servers are used to route call signaling among SIP terminals, by querying an internal database (DB). If the DB query gives no match for the current callee address, or if an error on the resulting next hop (SIP proxy) is obtained, the proxy releases the call and sends a *Not Found* message back to the caller. This fact may happen also and particularly in presence of a non-SIP called terminal; what really happens is that although the callee receives the SIP setup message, it is unable to generate an appropriate SIP response.

In order to forward call setup requests from a SIP based terminal (UA) to a H.323 user, a gateway entity (IWF) should include all the interworking functionality needed to translate transparently the SIP messages to H.323 signaling and vice versa.To make the correct forwarding of calls through the IWF possible, the client agents of the signaling servers (i.e. the SIP servers and the H.323 gatekeepers) should share some information about the presence of users behind the specific IWFs. Such information should be dynamically exchanged among the SIP servers, the IWF, and the gatekeepers. Although this action is not crucial between gatekeepers and gateways in an H.323 domain, there is not a straightforward solution for the SIP-to-IWF relationship, and a sort of specific protocol seems to be required.

A proposed solution for this issue makes use of the TRIP protocol. Another solution could be based on the adaptation of the SIP protocol and the change of SIP proxy functionality. However, both solutions seem to be not very suitable and won’t be followed.

A possible approach that could be used to solve this issue is the insertion of a new module (software or hardware) acting as a SIP proxy server. This module should forward all the calls coming from SIP terminals to both the next hop SIP server and the SIP-to-H.323 gateway (IWF). However, the main drawback of this approach is that it requires the duplication of call signaling for both SIP-to-SIP or SIP-to-H.323 calls. This solution is fast but it increases (duplicates) the signaling traffic sent through the IP network.

An improved solution could be obtained by starting a new call request at the SIP proxy server as soon as a “Not found” message or a “Time out” message has been received. The new calling process is performed towards the preconfigured IWF. This solution decreases the signaling traffic, but increases the call setup time (up to $2\cdot T_{out}$, where T_{out} is the time-out for the SIP call). To be noticed that the proposed solution is a compromise between the increase of signaling traffic and call setup delay.

3 The Network Architecture

To describe the network architecture, let us start observing what happens if in a pure SIP network, a SIP user addresses a call for a user that is in an H.323 network. We suppose that the caller and called users are respectively a SIP user, whose address is *sip:amalfi@a_sip.com*, and an H.323 user, whose address is *positano@b_h323.com.* The address of the gateway is: *sip:gw.a_sip.com.*

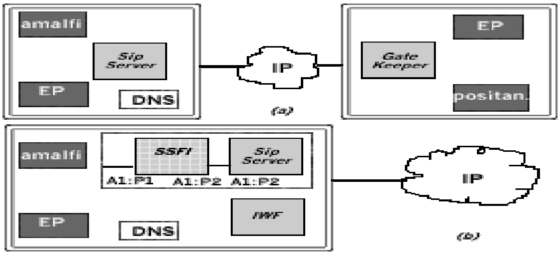


Fig. 1. (a) SIP-H323 standard interworking architecture(EP: end-point); (b) simplified structure of modified network

The user *amalfi* send a SIP INVITE message to the pre-configured SIP proxy server. As soon as the SIP Proxy Server receives the INVITE message, it tries to resolve the address contained in the field *To* of the message, consulting its “contact database” or by means of the DNS. If it cannot find any correspondence in the “contact database” for the user, it forwards an INVITE message to the *b_h323.com* domain; however, the H.323 domain cannot process successfully the message. Then the SIP Server answers the INVITE request by sending a *404Not Found* error message or a *408Request TimeOut* to the *amalfi* user. The previous result occurs even if an IWF is introduced in the SIP architecture (fig. 1). This is due to the fact that there is no mean to let the SIP server aware about the correct route of the SIP requests through the gateway. In other words, a call initiated by a SIP client and directed to a H.323 user would give negative results because of the fact that the SIP Server doesn’t know that it could address the call via the IWF. A possible solution proposed by the IETF is to register the IWF at the SIP Server using the TRIP protocol [4]. But even in this case it is necessary to have a new SIP Server in the network which is aware of the IWF and able to interpret the TRIP protocol.

By deeply examining the previously described scenario, it is possible to observe that the call failure towards an H.323 user is translated in a *404Not Found* or *408Request TimeOut* error message that is first received by the SIP server and then forwarded to the caller (*sip:amalfi@a_sip.com* in the previous example). Noticeably, when receiving these error messages, the server might guess that the called user belongs to the H.323 domain and try to forward the call to the IWF.

This consideration is the basis of our scenario in which a SIP call that cannot be forwarded to the called user within the SIP domain is relayed through the IWF to the H.323 domain. For this scope, a new software component has to be introduced, the *SSFI* (Sip –Server –Functional to Interworking). The SSFI can be seen as a very simple and stateless SIP proxy server that just forwards all incoming messages (both requests and responses). The only functionality that it implements is to look for *404Not Found* or *408Request TimeOut* error messages and, when one of these messages is received, to translate them in a *302Moved Temporarily* redirection message with the address of the IWF in the *contact* field. Any other message that will arrive to the SSFI, will be just forwarded to the client.

In order to minimize the impact on the original architecture, the new SSFI can be introduced simply by configuring the SIP terminals to let them use the SSFI as the default proxy. As an example, if a terminal uses as its default outbound proxy a SIP server at the address A1:P1 (Address:Port), when using the new SSFI module, the latter is configured to accept messages on A1:P1, while the original SIP server will accept messages at A2:P2. If SSFI should run on the same machine as the SIP proxy server, then A2≡A1 and P2 is one of the ports available on the server. Obviously the SSFI should forward every incoming call (from SIP user clients) towards the original SIP server using the socket A1:P2 (or A2:P2). We suppose that the SSFI and the SIP proxy are running on the same system.

The message flow between the SIP nodes is as follows: an INVITE message sent from the caller reaches the SSFI, the SSFI forwards it to the SIP server; if the SSFI doesn't receive a *200Ok* or *404NotFound* message within a time *t*, it starts trying to route the call towards the H.323 network by means of the IWF. We set this time *t* to $T_{out}/2$ (note that this isn't the optimal choice) [8].

4 Temporal Diagram

A client in the home SIP network sends an INVITE. The client asks *positano@b_h323.com* to establish a two-party conversation. The SSFI accepts the INVITE request and forwards the request to the SIP proxy server.

Both the SSFI and the Proxy Server set a Time-out counter. When the SIP proxy counter reaches the maximum value (T_{out}), the INVITE request is canceled. If the SIP proxy finds the called user before $T_{out}/2$, the terminal will send a *200 Ok message.*

If a *404 Notfound* message is sent to the SIP proxy before $T_{out}/2$ then the SSFI begins a new calling process in an other network using the gateway. The SSFI does not forward this response, but replies to the caller with the status codes *301* (*Moved Permanently*) or *302* (*Moved Temporarily*) specifying the IWF location with the Contact field. The caller then sends a new INVITE request to the SSFI with Request-URI set to the address specified in the Contact field.

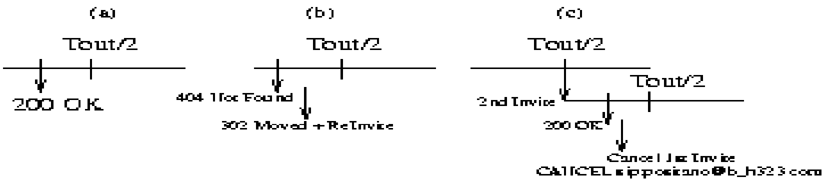


Fig. 2. **(a)** Successful transaction at SSFI; (b) Not Found in Sip network;(c) a successful response from the H.323 side, after the expiry of the first $T_{out}/2$

If no messages arrive to SSFI in a $T_{out}/2$ time, it starts a parallel search in the H.323 network. The SSFI then sends a new INVITE request to the SIP proxy with the same *To* (including tags), *From* (including tags), *Call-ID*, *Cseq* fields, but with a different Request-URI. Then it resets the Time-out counter. The Request-URI of the INVITE request is set to the IWF URI. For the SIP Proxy this request corresponds to a new transaction, and it should be proxied.

The "branch" parameter, in the new INVITE, is set to a different value. Actually this token must be unique for each distinct request. The SSFI uses the value of the "branch" parameter to match responses to the corresponding requests. CANCEL and ACK requests must have the same branch value as the corresponding requests they cancel or acknowledge. In this state, if a "*not found*" message arrives from the SIP network within $T_{out}/2$ seconds, the SSFI will keep on staying in a "wait" state. If a "not found" message arrives also from the IWF, SSFI will forward it to the SIP proxy, which will close the session. If a *200 OK* message arrives from one of the two networks, the SSFI will forward it as usual and, if necessary, will send a CANCEL message to the other network. The CANCEL message must be sent if a positive response arrives during the next $T_{out}/2$ seconds.

Just as an example, if we suppose that a *200 OK* response arrives from the IWF within the next $T_{out}/2$ seconds, the SSFI must send a CANCEL message to the SIP proxy. If neither the *200 OK* message nor the "not found" message should arrive from one of the two ways, the SIP proxy server will close the session, after 3/2 T_{out}.

We do note that, if a *200 OK* message arrives from the IWF, it is possible to update the DB of the SIP proxy server in order to route future calls addressed to the called user, directly to the IWF. The SSFI could make this updating, sending special REGISTER messages to the SIP proxy.

5 SSFI: State Machine

The idle state of SSFI is T (Transparent). When the SSFI receives an INVITE message, its state changes to S (SIP context), and its counter is set. In S state, when a *200 OK* arrives from the SIP network, the SSFI goes back into T state; otherwise, when a *404 Notfound* message arrives, the SSFI goes into H state (H.323 context).

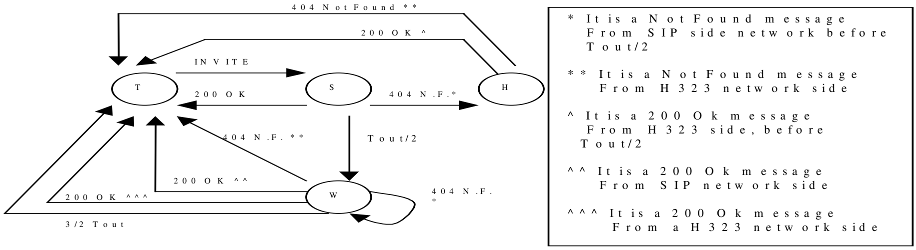

404 Not Found **
200 OK ^
INVITE
T
S
H
200 OK
404 N.F.*
404 N.F. **
Tout/2
200 OK ^^
200 OK ^^^
W
404 N.F. *
3/2 Tout
* It is a Not Found message From SIP side network before Tout/2
** It is a Not Found message From H323 network side
^ It is a 200 Ok message From H323 side, before Tout/2
^^ It is a 200 Ok message From SIP network side
^^^ It is a 200 Ok message From a H323 network side

Fig. 3. State Machine

In the H state, SSFI begins a new session sending a "*moved*" message to the caller. When either a *200 OK* or *404 Notfound* message is received the SSFI goes back to the idle state T. Furthermore when in S state, after $T_{out}/2$, SSFI reaches the W state. In this state (Waiting) if a *404 Notfound* message arrives from the SIP network, the SSFI continues waiting for some responses from the gateway.

6 Conclusions

In this paper the problem of the interworking between SIP and H.323 networks has been considered. The problem of call forwarding through different domains arises for calls generated from a SIP domain and directed to a H.323 domain. A possible simple solution has been proposed and described, taking into account particularly the problem of backward compatibility with previously installed SIP and H.323 networks and legacy systems. For this reason, the proposed solution does not use new protocols between signaling systems and does not require any modifications of SIP/H.323 terminals nor servers. The call can be forwarded to both domains in serial or parallel manner. A compromise is chosen in order to balance the generated signaling traffic and the call-setup delay.

References

[1] "Packet based multimedia communication systems", Recom. H.323 – ITU-T, Feb. 1998

[2] "Call Signaling Protocols and Media Stream Packetization for Packet Based Multimedia Communications System", Reccom. H.225.0, Version 2 - ITU-T, Feb. 1998

[3] "Control protocol for multimedia communication", Recom. H.245.0, ITU-T, Feb. 1998

[4] J. Rosemberg, , H Salma, "Usage of TRIP in Gateways for Exporting Phone Routes" March, 2000

[5] Singh, Schulzrinne, "Interworking Between SIP/SDP and H.323" May 12, 2000

[6] H. Agrawal, R. R. Roy, et Al. "SIP-H.323 Interworking Requirements", February 22, 2001

[7] H. Agrawal, R. R. Roy, et Al. "SIP-H.323 Interworking", July 13, 2001

[8] H. Schulzrinne, J. Rosemberg, et Al. "SIP: Session Initiation Protocol", July 20, 2001

High Router Flexibility and Performance by Combining Dedicated Lookup Hardware (IFT[1]), off the Shelf Switches and Linux

Christian Duret[1], Francis Rischette[1], Joël Lattmann[1], Valéry Laspreses[1], Pim Van Heuven[2], Steven Van den Berghe[2], and Piet Demeester[2]

[1] France Telecom R&D, Issy les Moulineaux, France
{christian.duret, francis.rischette, joel.lattmann, valery.laspreses}@francetelecom.com
[2] IMEC, Ghent, Belgium
{pim.vanheuven, svdberg, demeester}@intec.rug.ac.be

Abstract. In this paper we propose a new router architecture that combines both flexibility and performance. This router architecture aims at combining the best of two worlds: the commercial routers, which have a proven track for stability and performance but lack the flexibility of routers with open source operation system. The latter is particularly flexible because the source code is accessible for analysis and modification purposes as opposed to the traditional commercial routers, whose software can be altered by their manufacturers only.

1 Motivation and State-of-the-Art

The exponential growth of Internet traffic has yielded a dramatic development effort of the IP routers technology. Moreover, the deployment of value-added IP service offerings (ranging from a QoS-based access to the Internet to real-time services, like IP videoconferencing) has lead to an important development of specific capabilities (traffic conditioning, marking, scheduling and metering) that are supported by some - if not all - the routers of the Internet. The consequence of the activation of such enhanced capabilities is twofold: a demand for an increase of the routers' switching performances together with the availability of multi-functional and multi-service routers.

Other important concerns deal with IP security, multicast, and Virtual Private Networks services. Therefore, the IP routers that are exploited in a multi-service environment need to be flexible enough in order to address current and future requirements.

For the past decade, Linux has received considerable interest not only from the research community, but also from the industry. An extensive description of Linux features and related bibliography can be found in [1]. Recently, an implementation for DiffServ over MPLS [2] has been released by some of the authors of this paper.

[1] IP Fast Translator

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 1117-1122, 2002.

The main issue raised by the use of Linux-based routers deals with their switching and forwarding performances:

- They are bounded by the CPU and are difficult to predict since both the data and the control planes run on the same CPU;
- Another problem is the interrupt overhead. Note that alternatives exist which are based on polling [3].
- Even more important is that most of these routers are built around commodity PC, and therefore inherit of their shared bus limitations;

Commercial routers provide more than acceptable switching performances. Their main drawback is their lack of flexibility. Thus, whenever an IETF standard is not implemented yet, and/or some functionality is missing, it becomes necessary either to rely on the roadmap of a given manufacturer for the introduction of new features, or to add adaptation boxes, where it is feasible.

The commercial routers whose architecture is based upon a high performance CPU and interface cards linked together by a shared bus, are not sufficient anymore to keep pace of the constant increase of Internet traffic, hence overwhelming the Moore's law. A new class of components, dedicated to high speed network layer processing has emerged for about a year: the network processor. Unfortunately, network processors are clearly designed in an opposite way as the Linux paradigm.

2 The IFT-Based Experimental Router

Several years ago, FTR&D has developed a research program on high speed networking techniques to be initially deployed within an ATM context, so as to address the above-mentioned issues. One way to address the switching performances issue consists in system optimization. Looking at a conventional router, one can see that less than 5% of the system software runs in the data path, but is responsible for more than 95% of the execution time. Only a small part of the related functions has to be "wired" to reach the performance level that is needed today, this level being around 1.5 x 10^6 packets/second per Gigabit/s bit rate at the interface level. Among these functions, classification ("The process by which a data packet is examined and policy decision are made which affect down-stream processing" [4]) is a critical one, and it clearly requires as much flexibility as does a purely software-based implementation to handle forwarding decisions, filtering such as Access Control Lists (ACL), an increasing set of encapsulations headers, forthcoming protocols (IPv6)...

Generally speaking, the incoming frame is characterized by a set of fields within a succession of headers, whose respective contents could possibly be analyzed against a set of patterns. Each individual analysis is defined by the position of the field within the frame, the set of patterns against which to compare the content of the field, an action to be performed in the case of a match (either a link that leads to another field to be analyzed, or a final result that indicates where to send the packet, or a default treatment). Figure 1 below is an example of such behavior for basic IP forwarding.

The implemented lookup process is basically a "multibit Trie" allowing for either exact range or longest match. An extensive survey of lookup algorithms can be found in [5]. The complexities reported for this lookup scheme are:

Worst case lookup time O(W)

- Worst case update time O(W/K + 2K)
- Worst case memory size O(2kNW/K)

Where N is the number of entries, W the length of the address and K the size of the bit slice (or "stride" according to [5]).

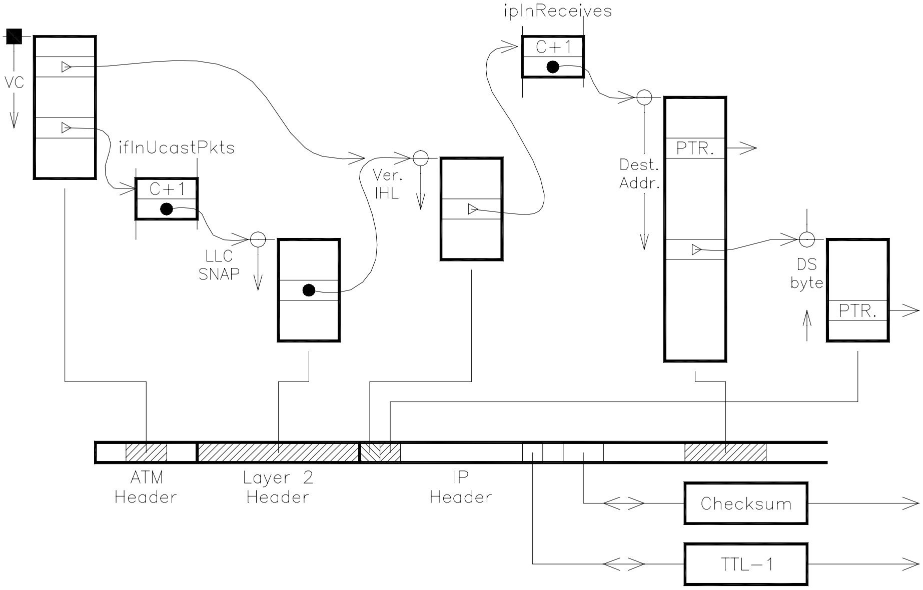


Fig. 1. Successive header fields to be processed for basic IPv4 forwarding. The shaded areas within the incoming frame are the header fields that are analyzed through the IFT. The sequence of the analyzed fields and related counters update is fully defined by the pattern store memory that implements a finite state machine, whose transitions are triggered by the incoming packet (upper part of the figure). A match may also trigger external processes to keep track of layer succession, check the header, update counters, update checksum, Time To Live (TTL) and Differentiated Service Code Point (DSCP). The IFT analysis result is mapped onto a VCI (Virtual Channel Identifier) value that implicitly designates the output port. These processes depicted in the lower part of the figure are protocol-dependent.

The worst case lookup time is 120ns for IPv4 addresses in the present hardware implementation, to be compared to 2.99us reported in [5] for a software implementation executed on a 200 MHz Pentium-Pro based computer under Linux.

By nature, there is neither layer nor any field restriction in the analysis: upper layers may be processed through linked tables. The worst-case lookup time is 345ns for a basic TCP-UDP/IP 5-tuple, to be added to regular forwarding process time. This classification is performed by implementing a "set-pruning trie" data structure according to the proposed taxonomy in a recent survey of algorithms for packet classification [6]. The properties of this structure are:

- Worst case lookup time O(dW)
- Worst case memory size O(dN)

Where d is the "dimension" of the classifier, that is to say the number of header fields of W bit length on which a number N of classification rules apply. The large amount of memory is due to the fact that some fields may need as much as dN tables to ensure that every matching rule for a given field will be traversed depending upon the result of the analysis of the previous field. No backtracking nor linear search are needed allowing to analyze each relevant field only once on the fly.

Backtracking, as implemented in "Grid-of-tries" [7], reduces the storage complexity to O(NdW) at the expense of O(Wd-1) for worst-case lookup time complexity.

Incoming packets are analyzed at line rate by reading the IFT control memory. A software driver running on the IFT host is in charge of writing it. This driver offers a set of updating functions: insertion and removal of patterns. It constantly provides the global consistency of the control memory, without the need of recurrent tables reorganization. The IFT driver runs on a logical copy of the IFT memory and performs incremental updates, thus the memory bandwidth required for update operations is several orders of magnitude lower than the bandwidth required by incoming packet processing.

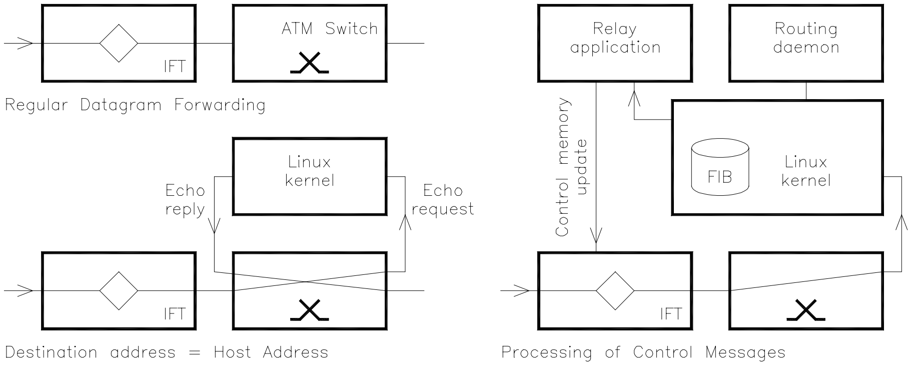


Fig. 2. Examples of packet processing.IFT-only functionality: The IFT runs a copy of the kernel Forwarding Information Base (FIB). A datagram whose destination address has been recognized is forwarded directly by the IFT to the switch fabric where it is forwarded to the output interface that leads to the next hop associated to the contents of the destination address field of the datagram.Linux control path functionality: a datagram destined to the router is forwarded to the router Linux host. The Linux kernel then processes this datagram. For example, if it is an Internet Control Message Protocol (ICMP), Echo Request message, then the kernel sends an Echo Reply message back to the originating host through the switch fabric.Linux control and IFT configuration functionality: a datagram destined to the router is forwarded towards the router Linux host. The Linux kernel sends this datagram up to the application layer. For example an Open Shortest Path First (OSPF), Link State Advertisement (LSA) packet is sent to the routing daemon. The daemon will update the kernel FIB if needed. The corresponding message is then copied in the IFT control memory.

The communication within the IFT-based experimental router is performed through an ATM switch fabric that directs IFT-processed packets towards external (most of packets) or internal interfaces for being handled by the Linux host and the control plane processes. Thus, aside the IFT driver, the role of the Linux host is threefold:

- In the data plane, processing of the datagrams that were sent to the Linux kernel by the IFT module (such as time exceeded ones or those containing options or directly addressed to the router);
- Running the control plane functionality;
- Configuring the IFT forwarding table through a user relay application.

The control path functionality is comparable to a regular Linux-based router. Figure 2 gives the three possible scenarios that can occur when a packet enters the experimental router. Routing protocol packets are an important example because these packets can update the routing table inside the Linux component. These changes have to be reflected in the IFT forwarding table too. This leads to the third role of the Linux components: the configuration tasks that consist of mapping the Traffic Classifier configuration commands and routing updates using netlink sockets [8] onto IFT header pattern entries. This has the advantage that software-based routing daemons can be re-used on the experimental platform without the need for any modifications.

3 Future Work

The present router design is based upon an ATM switch. Ongoing developments include the support of Fast and Gigabit Ethernet interfaces. The architecture described in this paper applies to a design based upon an Ethernet switch as well. In this case, the IFT analysis result, instead of being mapped to an ATM connection, is mapped onto a Medium Access Control (MAC) frame, whose Destination Address field is either a host, a gateway or the Linux host itself. Additionally, most of Gigabit Ethernet switches provide priority queuing mechanisms through the implementation of the IEEE 802.1p standard that may be useful for implementing Diffserv-based routing and QoS mechanisms.

The IFT developments have been considered for the implementation of a Multimedia Switch Router [9]. Security applications are also considered [10]. Another application of this kind of platform could be admission control facilities that would be based upon "on the fly" identification of elastic and streaming flows [11].

4 Conclusion

In this paper we explained that current marked trends push for both flexible and high performance routers. Current router options are either high performance (commercial routers) or flexible (open source-based PC routers).

As a solution to this problem, we propose a router architecture that consists of the combination of fast dedicated look-up hardware, off-the-shelf switches, and the Linux OS. The combination of these components provides:

- A performance level that can easily be compared to the switching performances of commercial routers;
- Scalability through the use of off-the-shelf switching fabric (currently ATM, Fast and Gigabit Ethernet later on);

- The flexibility at the control path equal to that of an open source PC router;
- The extensive developer support that have been engaged on Linux-based routers;
- A clear separation between forwarding and control planes.

Acknowledgements. This work was partly undertaken in the Information Society Technologies (IST) TEQUILA project, partially funded by the Commission of the European Union. Part of the work is also sponsored by the Flemish Government through two IWT scholarships. The authors would also like to thank the rest of the Tequila colleagues who have contributed to the ideas presented in this paper and Jacques Le Moal, Jean Louis Simon (France Telecom R&D) for their contribution to the IFT project.

References

[1] D. Griffin editor “D2.1: Selection of Simulators, Network Elements and Development Environment and Specification of Enhancements" http://www.ist-tequila.org

[2] Pim Van Heuven, Steven Van den Berghe, Tom Aernoudt, Piet Demeester, "RSVP-TE daemon for DiffServ over MPLS under Linux", http://dsmpls.atlantis.rug.ac.be

[3] Benjie Chen et. al.," The Click Modular Router Project", http://www.pdos.lcs.mit.edu/click/

[4] "Programming & Reprogramming: Keeping the speed without Losing your Mind" in Network Processor Summit - Networld+Interop 2000

[5] Miguel A. Ruiz-Sanchez, Ernst W. Biersack, Walid Dabbous "Survey and Taxonomy of IP Address Lookup Algorithms" in IEEE Network March/April 2001

[6] Pankaj Gupta, Nick McKeown "Algorithms for Packet Classification" in IEEE Network March/April 2001

[7] V. Srinivasan et al., "Fast and Scalable Layer four Switching" in Proc. ACM Sigcomm, Sept. 1998

[8] G. Dhandapani, A. Sundaresan “Netlink Sockets – Overview” http://qos.ittc.ukans.edu/netlink/html/

[9] Michel Accarion, Christophe Boscher, Christian Duret, Joël Lattmann "Extensive Packet Header Lookup at Gb/s Speed for an Application to IP/ATM multimedia switch router" In World Telecommunication Congress - International Switching Symposium, Birmingham May 2000

[10] Olivier Paul, Maryline Laurent, Sylvain Gombault, "A Full Bandwidth ATM Firewall" in Proc. of the 6th European Symposium on Research in Computer Security, Toulouse, France, October2000

[11] N. Benjameur, S. Ben Fredj, S. Ouslati-Boulahia, J. Roberts, "Integrated Admission Control for Streaming and Elastic Traffic" in M. Smirnov, J. Crowcroft, J. Roberts, F. Boavida (Eds), "Quality of Future Internet Services", Springer, LNCS 2156, 2001.

Group Security Policy Management for IP Multicast and Group Security

Thomas Hardjono[1]and Hugh Harney[2]

[1] VeriSign Inc., 401 Edgewater Place, Suite 280,
Wakefield, MA 01880, USA
thardjono@verisign.com

[2] Sparta Inc., Secure Systems Engineering Division,
9861 Broken Land Parkway, Suite 300,
Columbia, MD 21046, USA
hh@columbia.sparta.com

Abstract. The current work focuses on the area of group security policy within secure IP multicast and secure group communications. The work explains the background and context, introduces a Group Security Policy Framework, and describes how this fits within the broader Multicast Security Framework developed within the IETF. Finally, the current status of developments within group security policy in the IETF is discussed.

1 Introduction

Group communications, also commonly called multicast, refers to communications in a group where the messages can be sent by any member and are received by all members. They range from mailing lists to conference calls to IP Multicasting. Often the need for data protection arises, which requires the group to handle the messages in a consistently secure manner. To accomplish this, cryptographic mechanisms and security policy must be shared and supported by the group as a whole. Because of this, special problems arise in managing the cryptographic and policy material as it changes or as the group changes.

The current work discusses the need for policies and policy-management for secure groups, placing the discussion in the context of the SMuG/MSEC Framework for Multicast Security in the IETF. The work described the Multicast Security Framework and identified the entities and interactions involved in group security policy management. It then focuses on a framework for group policy management for secure-groups, and explains the current status of developments in the IETF.

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 1123-1128, 2002.

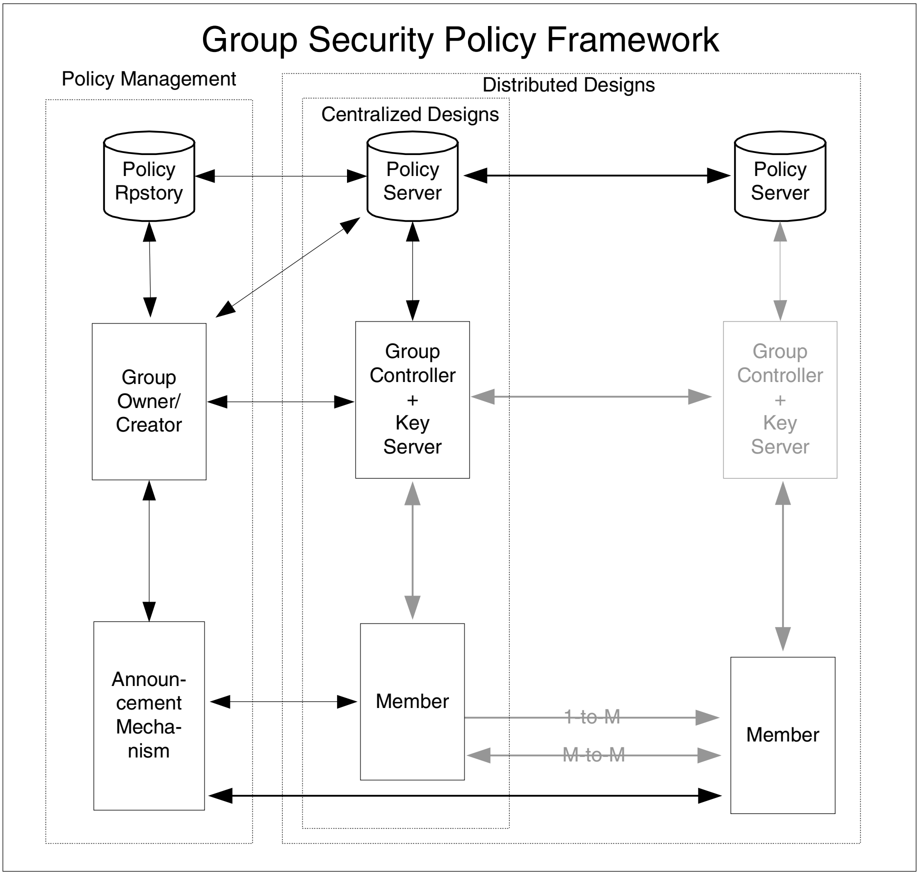


Fig. 1. Group Security Policy Framework

2 Group Security: Background & Framework

There is significant interest in the networking industry and content delivery network (CDN) industry to use IP multicast a vehicle for data delivery to a large audience. One major hindrance to the successful deployment of IP multicast and other group-oriented communication protocols has been the lack of security for both the content and the content-delivery infrastructure.

To this end, the IETF designated in mid-1998 the creation of the Secure Multicast Group (SMuG) under the umbrella of the Internet Research Task Force (IRTF) to research and develop protocols for multicast security. This IRTF group has since been formalized into a IETF Working Group, called Multicast Security (MSEC), early in 2001. The architecture and designs developed within SMuG have largely been carried-over into the MSEC WG with the aim of further refining and formalizing into specifications for a set of standards documents (RFCs).

The Secure IP Multicast Framework and Building Blocks document [HCBD00] of the IETF describes a number of entities, which participate in the creation, maintenance, and removal of secure multicast groups. Those that are of concern for group security policy are the *Group Controller and Key Server* (GCKS), the *Group Policy Server* (GPS) and *Member* (Receiver and Sender).

The Framework of [HCBD00] identified three broad problem-areas that need to be addressed. These are group key management, data/content handling (i.e. treatment of messages in a crypto context) and group policy. It is the later problem area that is of interest here, and will be further discussed in the following sections.

3 Group Security Policy Framework

The intent of the Framework of [HCBD00] is to present a high-level roadmap for the development of technologies that implement group and multicast security. Thus, to that extent, it was intended that each problem-area would develop its specific or focused framework or architecture. An example of a more focused architecture is one for group key management as reported in [HBH00, BCD01]. In the following section, we discuss a framework for group security policy, using the Framework of [HCBD00] as the starting point. Figure 1 shows a framework for group security policy where additional entities (over those in [HCBD00]) have been introduced relating to group policy. Both centralized and distributed designs are still shown, though slightly skewed to emphasize the distributed designs involving the policy-related entities.

3.1 Group Owner/Creator (GOC)

The Group Owner/Creator (GOC) represents the entity that is understood by all participants and entities in the network as the ultimate controller of the secure group. The entity is understood as having among others the following tasks:

- *Defining group policy*:
  The GOC defines all types and levels of policies pertaining to the group. This assumes that the network infrastructure for policy creation and assignment exists and can be deployed.
- *Setting-up network services*:
  As the creator/owner of a group, the GOC is assumed to also have network resources at all necessary layers of the network to enable the running of the group.
- *Defining membership*:
  The GOC defines the constituency of the group which it is setting-up. The basis of the membership of the group can be loose or tight, using host/user identity, IP addresses, certificates, or even a predefined access control list.
- *Sending out announcements/invitations*:

The GOC is also responsible for putting out an announcement or call to join through the mechanisms it selects. This could be using IP broadcast or multicast, advertising on a website or other mechanisms.

- *Terminating groups*:
  The GOC is also responsible for concluding a secure group, particularly if that group consumes (network) resources.

3.2 Group Policy Servers (GPS)

The Group Policy Server represents the entity that holds available the policies pertaining to groups. This information can be split into the policy items available for the general public (of non-members) and those available only to designated members of a group.

- *Publicly available policy items*:
  This is information pertaining to a secure group that has been previously announced through some public medium and which can be used by hosts/users to evaluate their eligibility to join a group.
- *Private policy items*:
  This is information that is only available to entities that have passed the membership eligibility test. The policy items may represents additional group-related policies that a (strongly authenticated) member needs to know in order to proceed further with participating in the group.

3.3 Group Policy Repository (GPR)

The Group Policy Repository (GPR) has the function of storing the secure group policies, each with the suitable protection levels and with access to it subject to appropriate authorization. Typically, authorization to access the GPR is provided only to the Group Owner/Creator (read/write/modify) and to the GCKS and Policy Servers (read). The first aim of the GPR is to make the policies pertaining to secure groups available on-line. The same is true for GCKSs. The second aim of the GPR is to allow dynamic update of policies by the Group Owner/Creator in cases when updating some policies does not endanger a group in progress.

3.4 Group Policy Announcement Mechanisms

The Group Policy Announcement (GPA) is a functionality that is aimed at making available information about groups to the intended recipient of such announcements. In the case of a Closed Secure Group, the announcement's intended recipients would be the members pre-selected by the Group Owner/Creator. In the case of an Open Secure Groups, the announcement will be readable by the public.

4 Group Security Policy Token

Current work in the IETF have so far focused more on how to define and represent the *security mechanisms* policies in the context of IP multicast security, where IP multicast is seen as the primary transport for group-oriented communications. The *Group Security Policy Token* (GSPT) [HHMCD01] is a structure that represents security mechanisms (and their parameters) used within a secure group (Figure 2). Not all elements of a GSPT for an instance of a group are made public through the announcement. The work of [HHMCD01] is a continuation of earlier work on group policies within the framework of GSAKMP [HCHMF01]. The elements of a GSPT (or *categories* in [MHCPD00]) specify the policies that are to be followed by members of a group, and consist of the following:

- *Policy Identification*: A group must have some means by which it can identify an instance of Group Security Policy in an unambiguous manner.
- *Authorization for Group Actions*: A Group Security Policy must identify the entities allowed to perform actions that affect group members.
- *Access Control to Group Information*: Access control policy defines the entities that will have authorization to hold the key protecting the group data.
- *Mechanisms for Group Security Services*: Identification of the security services used to support group communication is required. For example, policy must state the algorithms used to derive session keys and the types of data transforms to be applied to the group content.
- *Verification of Group Security Policy*: Each policy must present evidence of its validity.

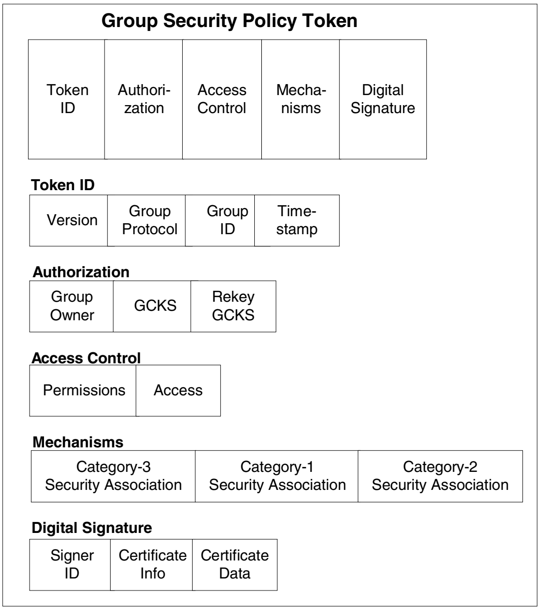


Fig. 2. GSPT Structure

5 Remarks and Conclusion

The current short paper has discussed the need for policies and policy-management for secure groups, placing the discussion in the context of the SMuG/MSEC Framework for Multicast Security in the IETF. The work then presented a more policy-focused framework/architecture using these existing entities, while introducing others that are relevant to group security policy management. The Group Security Policy Token (GSPT) was then presented and discussed. The GSPT represents the current status of development in the IETF in the MSEC Working Group with respect to group security policy.

References

[BCD01] M. Baugher, R. Canetti, L. Dondeti, *Group Key Management Architecture*, draft-ietf-msec-gkmarch-00.txt, June 2001, Work in Progress.

[HBH00] H. Harney, M. Baugher, T. Hardjono, *GKM Building Block: Group Security Association (GSA) Definition*, draft-irtf-smug-gkmbb-gsadef-01.txt, September 2000, Work in Progress.

[HHMCD01] T. Hardjono, H. Harney, P. McDaniel, A. Colgrove, P. Disnmore, *Group Security Policy Token*, draft-ietf-msec-gspt-00.txt, IETF, Work in Progress, Sept 2001.

[HCBD00] T. Hardjono, R. Canetti, M. Baugher, P. Dinsmore, *Secure IP Multicast: Problem Areas, Framework and Building Blocks*, draft-irtf-smug-framework-01.txt, September 2000, Work in Progress.

[HCD00] T. Hardjono, B. Cain, N. Doraswamy, *A Framework for Group Key Management for Multicast Security*, draft-ietf-ipsec-gkmframework-03.txt, August 2000, Work in Progress.

[HCHMF01] H Harney, A Colegrove, E Harder, U Meth, R Fleischer, *Group Secure Association Key Management Protocol (GSAKMP)*, draft-ietf-msec-gsakmp-sec-02.txt, December 2001, Work in Progress.

[MHCPD00] P. McDaniel, H. Harney, A. Colgrove, A. Prakash, P. Dinsmore, *Multicast Security Policy Requirements and Building Blocks*, draft-irtf-smug-polreq-00.txt, November 2000, Work in Progress.

[SMuG-MSEC01] `www.securemulticast.org`

Issues in Internet Radio

Yasushi Ichikawa, Kensuke Arakawa, Keisuke Wano, and Yuko Murayama

Dept. of Software and Information Science, Iwate Prefectural University
152-52 Sugo, Takizawa, Takizawa-mura, Iwate,Japan
{ichikawa,araken}@comm.soft.iwate-pu.ac.jp,
g031x169@edu.soft.iwate-pu.ac.jp, murayama@iwate-pu.ac.jp

Abstract. The World-wide Web(WWW) works now as the infrastructure over the Internet for multimedia applications. Internet radio is one of those applications and its growth is explosive. We have started operating an Internet Radio station with streaming music services since April 2000. An Internet radio can broadcast music over the network regardless of such geographic restrictions as the traditional radio systems have. There are some problems and services due to the Internet. This paper reports our operation as well as the issues. We propose our idea on some novel radio services as well.

1 Introduction

During the 80's the question was for what exact applications the Internet would be used best. We now know that the answer is WWW. The Internet has grown dramatically since WWW was introduced in the end of the 80's. Indeed, WWW is considered now as the infrastructure over the Internet for multimedia applications.

Internet radio is one of those applications. The growth of the number of Internet radio stations is explosive. There are more than 5000 stations operating over the Internet. The number of Internet radio stations has been increasing about 1000 stations each year.

The purpose of our research is to identify the issues to be dealt with by Internet Radio. This paper reports our initial effort to set up a radio station as well as its operation for several months. We describe issues and present an idea of some novel radio services.

2 Internet Radio Systems

The Internet radio stations are classified into two types according to their operations; commercial ones and private ones. The private stations operate differently from the commercial ones including the traditional radio broadcast stations. The private ones would select the music more from the service provider's viewpoint, whereas commercial ones need to provide the music favored possibly by many listeners. A famous commercial station would keep having more than 200 listeners.

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 1129–1134, 2002.

An Internet radio system has a client-server structure. A radio station has a server, and a user needs to have a client system such as an MP3 player. The multimedia authoring tools and the Internet have made it possible for us to set up private radio stations easily.

There are two types of music streaming services available on the Internet. One is to download music data and then play it. The other is to play music on music streaming server on a real-time basis. Our radio station uses the latter.

There are three systems available for setting up an Internet radio system, viz. The *Real system* [2], *Shoutcast* [3], and *Icecast* [4]. *Real System* is a server for a specific client system, the *Realplayer*. Most of the Internet radio stations use *Real System. Icecast* and *Shoutcast* provide MP3 streaming servers. MP3 is an MPEG Audio Layer 3, a compression format [5].

3 Flip over Radio(FOR)

In this research we set up our own radio station on the Internet called "Flip Over Radio (FOR)." At the moment we operate FOR on an experimental and private basis. We broadcast Indie music which is made originally by unknown artists who work independently from record companies. They have a limited opportunity in publishing their music such that the listeners can obtain the information only from specific magazines and music stores in Japan.

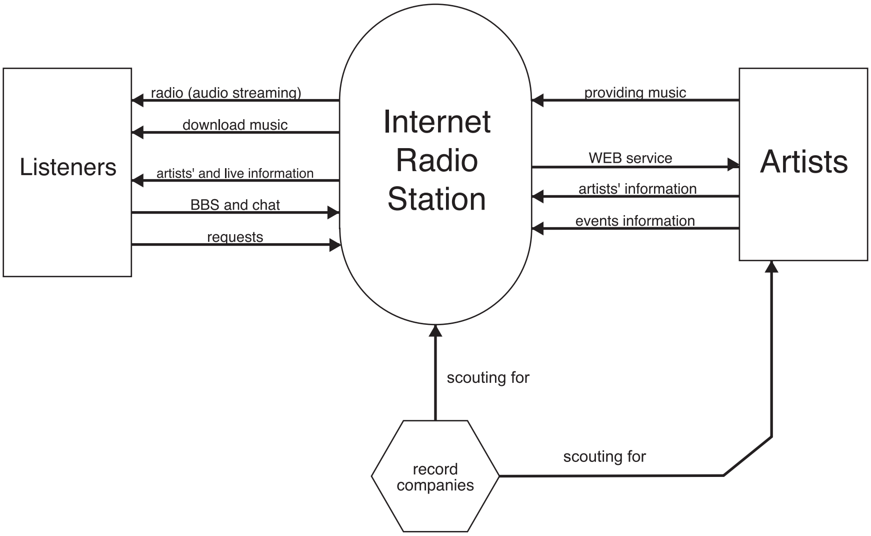


Fig. 1. The operational model of FOR

Fig. 1 shows the model of our radio operation. Our Internet radio station provide such an opportunity for both artists and listeners to exchange the information on music and artists. Our radio site is a media for this exchange. The artists provide the music that they composed and played as well as the related information. We provide them with tools such as the one to make their home page as well as the message board so that they can communicate with the listeners. Commercial promoters could make use of the information we provide to find a new artist and music, so that an artist could have an opportunity to get a commercial contract.

Table 1 shows the configuration of the server system of FOR.

Table 1. The server system of FOR

CPU	AMD K6-2 400MHz (Over DriveProcessor)
MEMORY	48M
OS	Laser 5 Linux 6. 0
HTTPserver	Apache
Streaming Application	Icecast [4] Shout [4] Icedj [6] Liveice [7]

Icecast is used to broadcast music. *Shout* selects a music to broadcast, and passes the music data to *Icecast. Liveice* is a real-time re-encoder and passes the encoded data to *Icecast.* We can mix several MP3 streams and audio inputs from mic(microphone) and *Liveice. Icedj* is used to run an *Icecast* radio station such that broadcasts a music at a certain time as scheduled in a program. It can be used together with *Icecast* to show the information on music being broadcast on the radio station's WWW page.

4 The Operation Report

We have been operating FOR since April 2000. Fig. 2 shows the number of total user access per month. We have not had so many users, presumably it is not because of Indie music, but due to poor amount of contents. Users would not listen to an Internet radio station if it broadcast the same songs repeatedly.

During July and August in 2001, we revised *icecast* in the latest version, so that the facility of the registration function started working well, which registers our radio server to the access ranking server on the Internet. 3 percent of connections were from our university, and 20 percent of connections were from Japan.

We found two requirements. One was that a user needs an easy-to-use interface. The other was that a lot of contents are required. We may well need a user interface in JAVA Applet, so that the software is installed automatically. A station with poor contents would not have the users who would visit the radio station site repeatedly.

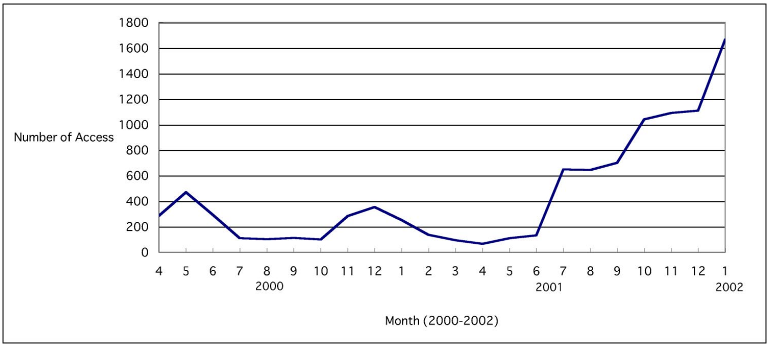


Fig. 2. User access per month (2000 - 2002)

5 User's Private Channel

We are planning to provide users with private channels, so that users can listen to their favorite music. This novel type of service is only possible on the Internet, but not on the traditional radio systems.

Fig. 3 shows the design of a private channel. The private channel operations are as follows :

1. A user registers his/her desirable channel ID and favorite music information on the web page and receives from the server a URL.
2. The registration process sends the registered information to a database engine which selects music. The database engine makes the play list of the user's favorite music.
3. The channel making process makes the user's private channel with the channel ID and the play list. The user makes access to the URL and listen to the music with the MP3 player.

We have implemented the first part, and half of the third part, from the above list. For the second part, we are planning to construct the music database with some attributes such as quiet and noisy, so that user's favorite music tunes can be selected automatically according to the user's taste.

There is a problems with this service. If we provide users with private channels on demand, we will require to run as many private channel processes as the number of user requests. The more private channel processes we have, the more loads the server gets and the slower the system operation becomes. We may need to explore the tradeoff between the performance of the server and the number of private channels. We may well need dynamic channel management.

6 Distributed Streaming

If we provide our radio service from only one site, the server site will be a bottle neck as the number of users increases. We propose distributed streaming

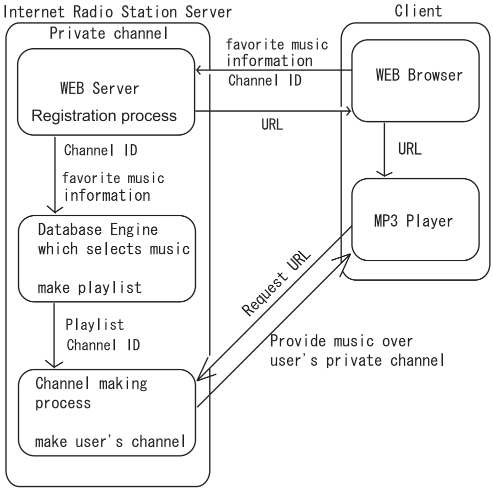


Fig. 3. The design of a private channel

by setting up relay servers. A relay server is an application level router which forwards MP3 data stream. A radio station transmits single music data to a relay server. A user connects to the nearest relay server. The relay server makes copies of the music data and sends them down to users. This operation looks similar to multicast as in Resource ReSerVation Protocol (RSVP)[8] whose multicast function operates at the network level.

Content Delivery and Distribution Networks(CDNs) may be one of a possible tool for this[9], CDNs provide users with an access to one of the distributed servers in the different locations over the Internet. The servers have a cache of an original content. A user has an access to the nearest CDNs server. There are many products and services of CDNs. We need further investigation on the use of CDNs for the Internet radio service.

We plan to provide users with private channels by making use of CDNs according to user's taste from the user's nearest relay server. Firstly, the relay server caches the music contents of the original server. Secondly, the private channel server makes a play list according to a request from a user, and sends it to the user's nearest relay server. Thirdly, its relay server makes channel and provides music according to users' taste with the play list.

7 Related Work

Most of the radio stations are operating on a commercial basis.

Among them the following site is one of those that have many services and users: http://www.netradio.com. It has more than 100 channels. They are classified firstly into global categories such as pop, rock, and so on. In a category, the channels are classified further into subcategories such as chronological groups.

The commercial sites provide users with a shopping function as well so that users can purchase CDs of their favorite music.

Another radio station: http://www.wolffm.com. deals with the various types of streaming such as MP3 streaming *Realplayer*, and *Windows Media Player*.

Our radio station provides only MP3 streaming and 32kbps bit-rate at the moment, however, we are planning to provide some other bit-rates of MP3 as well in future. Our radio station is managed on a private and non-profitable basis with one channel, the contents of 150 music tunes, and some artists information at the moment. We are providing only with the specific type of copyright-free music and the information on the almost unknown artists in Japan.

8 Conclusion

This paper introduced the Internet radio from the viewpoint of a service provider. Internet radio systems have many different properties from the traditional radio system. For example, a cultural revolution could be possible in music, since any type of music could be delivered over the Internet and some of them would never appear on the traditional commercial media.

We reported on the operation of our Internet radio station and identify some issues to be dealt with in future. The issues include providing users with their private channels and making the delivery system distributed.

Future work includes examining the database engine function of the private channel, making the delivery service distributed, and providing an easier user interface. We plan to implement those required functions into a client system using JAVA Applet with Java Media Frame(JMF)[10], provided in the Multimedia library of JAVA.

References

1. Flip Over Radio : http://radio.comm.soft.iwate-pu.ac.jp [***]
2. Real system : http://www.realnetworks.com/ [***]
3. Shoutcast : http://www.shoutcast.com [***]
4. Icecast : http://www.icecast.org [***]
5. Fraunhofer Research : http://www.iis.fhg.de/amm/ [***]
6. Icedj : http://www.remixradio.com/icedj/ [***]
7. Liveice : http://star.arm.ac.uk/˜spm/software/liveice.html [***]
8. L. Zhang, S. Deering, D. Estrin, S. Shenker, and D. Zappala: RSVP: A New Resource ReSerVation Protocol, IEEE Network Vol.7 Issue 5 pp.8-18 (Sep. 1993)
9. Balachander Krishnamurthy, Craig Wills and Yin Zhang, On the Use and Performance of Content Distribution Networks, ACM SIGCOMM Internet Measurement Workshop 2001
10. JMF : http://www.java.sun.com/products/jave-media/index.html [***]

[***] last access : Feb. 27, 2002

I/O Bus Usage Control in PC-Based Software Routers[†]

Oscar-Iván Lepe-Aldama and Jorge García-Vidal

Department of Computer Architecture, Universitat Politècnica de Catalunya
c/ Jordi Girona 1-3, D6-116, 08034 Barcelona, Spain
{oscar,jorge}@ac.upc.es

Abstract. This paper presents a performance analysis of a fair sharing mechanism for PC-based software routers, required when the I/O bus and not the CPU is the bottleneck. The mechanism involves changes to the OS kernel and assumes the existence of certain NIC functions, but does not require any changes to the PC hardware architecture.

1 Introduction

We can define a software router as a computer that executes a program capable of forwarding IP datagrams among network interface cards (NIC) attached to its I/O bus. It is well known that software routers have performance limitations. However due to the ease with which they can be programmed for supporting new functionality software routers are still important at the edge of the Internet. After this, the question of how to optimize software routers performance arises. In addition, if we want to provide QoS guarantees for traffic going through the router, we must find a suitable way of sharing resources. In other pieces of work the problem of fairly sharing router resources is tackled in terms of protecting [1,4] or sharing [6] the use of the CPU amongst different packets or data flows. However, the increase in CPU speed in relation to that of the I/O bus makes it easy for this bus to constitute a bottleneck, which is why we address this problem.

This paper presents our proposal for a resource sharing mechanism that allows QoS levels to be guaranteed in software routers by jointly controlling I/O bus activity and CPU operation. It is a software mechanism that does not require changes to the PC hardware architecture and which introduces low overhead and avoids intrusion. It requires that NICs dispose of several direct memory access (DMA) channels—one for each traffic flow—working independently and having a set of descriptors that store usage information—NIC's buffer occupancy or the total number of arrivals to the channel. Moreover, this paper presents a study of the properties of the mechanism, when considered in isolation, and a system performance evaluation, when the mechanism is incorporated into a software router. We will concentrate on software routers built on desktop PCs running general purpose, open source operating systems—FreeBSD, which implement networking functions within the kernel.

[†] This piece of work was supported in part by the Mexican Government through CICESE Research Center's and CONACyT's grants. Also supported by the Spanish Minister of Science and Technology through project TIC2001-0956-C04-01

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 1135-1140, 2002.

2 A Mechanism for Implementing I/O Bus Sharing

The mechanism we propose for implementing I/O bus sharing, and that we call Bus Utilization Guard (BUG), manipulates the vacancy space of the message buffer reception input queue of each DMA channel, so the overall activity at the I/O bus follows a schedule similar to one produced by a WFQ server. (For now on we referred to the I/O bus simply as the bus, and to a MBUF queue simply as a queue.) For minimizing intrusion, the mechanism is activated each T cycles and it is executed either by the CPU or by a suitable coprocessor placed at the AGP connector. For reducing overhead, the mechanism uses a two state behavior, monitoring and enforcing.

Assume that the mechanism is in monitoring state at cycle $k \bullet T$. Then, the mechanism gathers $D_{i,k}$—number of bytes transferred through the bus during period $((k-1) \bullet T,\ k \bullet T)$ by channel i. If $sum(D_{i,k}) < T/\beta_{BUS}$, where β_{BUS} is the cost per bit of bus transfer, the mechanism remains at monitoring state and no further actions are taken. On the contrary, the mechanism detects the start of a busy period and enters enforcing state. When at this state, the mechanism polls each NIC to gather $N_{i,k}$—number of bytes stored at the NIC associated with channel i—and computes the amount of bus utilization granted to each channel, or γ_{ik}, after the outputs of an emulated general processor sharing (GPS) server [5] with batched arrivals, or $G_{i,k}$. The input for the emulated GPS are the $N_{i,k}$ at the start of the busy period. Afterwards, the inputs are the amount of arrived traffic during the last period or $A,_{ik}=N_{i,k}-N_{i,k-1}+D_{i,k}$. BUG is work-conservative and thus

$$\gamma_{i,k} = G_{i,k} + (T/\beta_{BUS} - (G_{1,k}+\ldots+G_{N,k})\ ) \qquad (1)$$

Observe that $sum(\gamma_{i,k}) = T/\beta_{BUS}$, a situation that can lead to an unfair share. Consequently, BUG is prepared with an unfairness-counterbalancing algorithm. This algorithm computes an unfairness level per channel and if it detects at least one deprived flow, then it reduces $\gamma_{i,k}$ of every depriver flow by the corresponding unfairness value. One problem with this approach is that if unfairness is detected then

$$(\gamma_{1k} + \ldots + \gamma_{Nk}) \ /\ \beta_{BUS} = T \qquad (2)$$

That is, the unfairness-counterbalancing algorithm may artificially produce some bus idle time. This problem also arises when packetzing bus utilization grants, as shortly explained. Happily, a single mechanism, one that allows BUG to vary the length of its activation period, solves both problems. The length T of BUG's activation period, in general, keeps no relationship with any packet bus-transmission time—besides having to be at least larger than the largest. Consequently, when packetizing utilization grants it may happened that $mod(\gamma_{i,k}\ ,\ L_i)?\ 0$, where L_i is the mean packet length for channel i. Hence, some rounding off is required. We have tested rounding off both down and up and both produce particular problems. However, the former gave us a more stable mechanism. If nothing else is done, some bus idle time is artificially produced and the overall share assigned to that flow would be much less of what it should be. This problem can be solved if we let BUG reduced its next activation period length by some dt time value, where dt is the time due to rounding off. Evidently, this increases BUG's overhead. But as long as *dt* is a small fraction of *T*, the increase will remain at acceptable levels.

BUG will switch from enforcing to monitoring state, resetting the emulated GPS, any time that $sum(D_{i,k}) < T/\beta_{BUS}$.

3 BUG's Dynamics

We devised a series of simulation experiments to assess the performance of a PCI bus controlled by a BUG. For all experiments we compared the responses of three simulated buses: a plain PCI, a WFQ bus and a BUG regulated PCI. We are approximating the PCI operation by a Round Robin scheduler. Operational parameters where computed after a 33 MHz, 32-bit bus. Besides, we set queue spaces to infinity and set BUG's nominal activation period to 0.1 ms. Traffic load for all experiments was composed of three packet-flows soliciting each 1/3 of router resources. Flows differentiate themselves by the size of their packets: small (172 bytes), medium (558 bytes) and large (1432 bytes). Different experiments used different inter-arrival processes to show particular behavior.

In Fig.4.a we show responses to unbalanced constant bit rate traffic. Each line at every chart denotes the running sum of output bytes over time. The traffic pattern is as follows. At time zero, flow 1 and flow 2 start loading the system with a load level equivalent to 50% of a PCI bus capacity each; that is, 528 Mbps. Two ms later (first arrow; 20T = 2 ms) flow 3 starts loading the system also at 528 Mbps. Then, 2 ms later (second arrow) flow 3 multiplies its bit rate by four. From the first chart we can see that the ideal bus behavior allows a 50% bus share between flow 1 and 2 during the first 2 ms. Then, after flow 3 gets active, it allows a 33% bus share irrespectively of the load level of flow 3. From the second chart we can see that a plain PCI bus only adequately follows the ideal behavior during the first 2 ms—first arrow. Then, the round robin scheduling deprives flow 1 in favor of flow 3. Moreover, although flow 2 is lightly affected it also receives more than its solicited share. After time 4 ms—second arrow—all gets worst. From the third we can see that the BUG equipped bus behaves very much like the ideal bus does. Observe that when flow 3 gets on, the reactive nature of BUG is reflected. For the first two activation periods, or so, flow 3 gets bus use-grants above its solicited share, depriving the other flows. But then, BUG adjusts and before 1 ms has passed all flows start receiving their solicited share. Before time 10 ms, flow 1 starts lagging a little behind flow 2. This is due to rounding off mismatches. By algorithm definition, when this mismatch accumulates to a whole packet BUG will allow flow 1 to catch up. We have practiced more experiments like the above varying the order of the flows and the length and size of the load changes and we have always found congruent results.

In Fig.4.b and Fig.4.c we analyze the dynamic behavior of BUG under highly variant random load. For this pair of experiments each packet flow was run by an on-off source. On-state period lengths were set to a constant value. Packet inter-arrival processes were Poisson with mean bit rate equal to 3520 Mbps, or 300% of the PCI bus capacity. Off-state period lengths were drawn after an exponential random process with mean value set to 9 times the on-state period-length. Consequently, all flows overall mean bit rates were equal to 30% of the PCI bus capacity or 352 Mbps. Besides observing the system response to this kind of traffic, with these experiments we wanted to see if we could find any BUG pathology related to operating-mode cycles, where the continuous but random path into and out of enforcing mode may produce some wrong behavior. Consequently, we ran several experiments with different on-off cycle lengths. Here we present results for an on-state period-length 8 times the BUG activation period T (Fig.4.b) and for one of 0.5T (Fig.4.c). In both these figures, each chart left to right separately compares for each flow (flows 1, 2 and 3) the resulting output processes for each of the considered buses. Each line denotes the running sum of output bytes over time, and thus horizontal segments correspond to off-state periods. For reference, each chart also draws, as a running sum over time, the corresponding flow's input process. From both figures we can see that despite the traffic's fluctuations BUG quite well follows the ideal WFQ policy, while the PCI

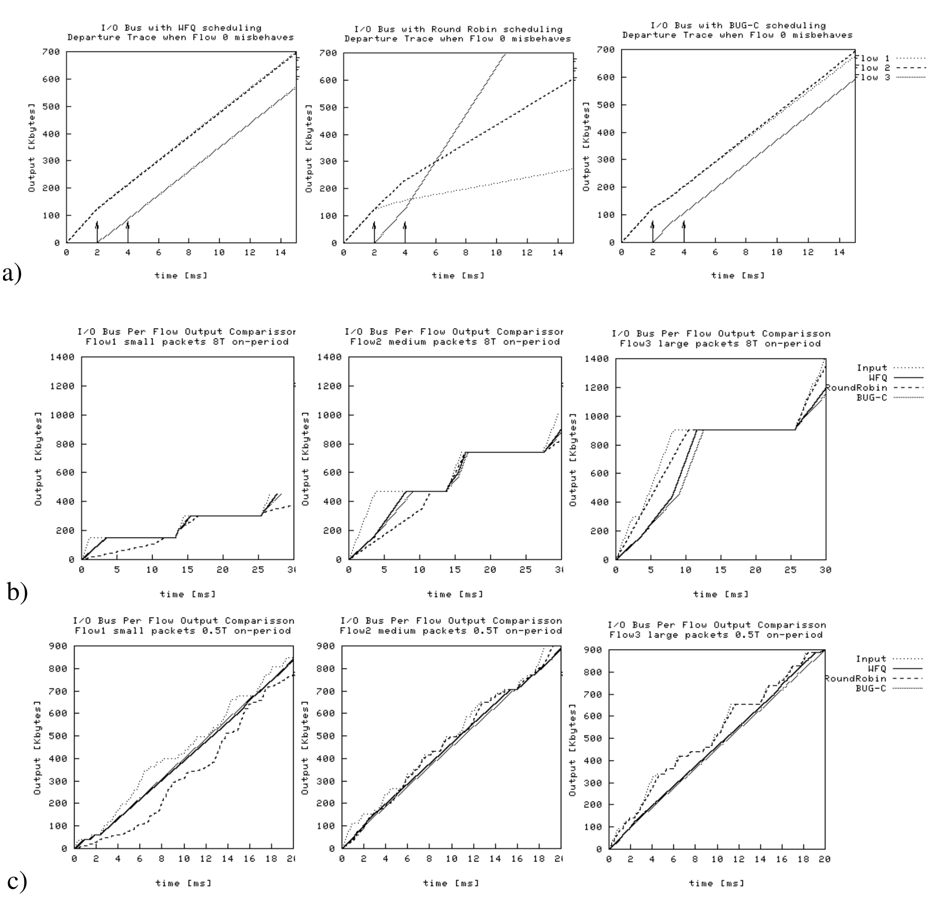


Fig. 1. Simulation results from BUG dynamics contrasting study under (a) unbalanced CBR traffic and (b,c) random and highly variable traffic. BUG's behavior is contrasted to the behavior of the ideal WFQ policy and the behavior of a PCI bus (approximated by a round-robin policy). Note that each chart at (a) contrasts the output processes of the three traffic flows described in the main text for a particular scheduling policy. While at (b,c) each chart contrasts the output processes produced by the three scheduling polices for one traffic flow.

like Round Robin policy again favors the largest-packet flow and affects the most to the smallest-packet flow. Of particular interest is what Fig.4.c show to us about BUG behavior. It seams that BUG is not macroscopically sensitive to a traffic pattern that repeatedly takes it in and out of enforcing mode.

4 System Performance Study

Here we study the performance of a PC based software router whose PCI bus in regulated by BUG. Operational parameters for the queuing network model were determined using software profiling, as described in [2]. The target system had a 600 MHz Pentium III CPU, a 100 MHz system bus, 10 ns EDO RAM chips and a 33 MHz, 32-bit PCI I/O bus. Software wise, the system was power by FreeBSD 4.1.1. Measurements were not taken for the bus service times. Instead, we used the description of the system operation [2]. We assume that data phases are of 1 cycle and that frame transfer is never pre-empted. We have considered Poisson traffic as input traffic, and which has a three-flow configuration as for the previous section.

We have performed the simulation with systems configured with two different CPUs. CPU1 works at 1 GHz and CPU2 works at 3 GHz. The system's I/O bus works at 33 MHz and has a 32-bit data path. Note that for the considered traffic, the CPU is the bottleneck for the system with CPU1 while the I/O bus is bottleneck for system with CPU2.

In Fig.5.a we show results for the basic software router. The left chart shows aggregated throughputs for offered loads in the range of [0, 1400 Mbps]. The other two charts show the share obtained for each traffic flow, firstly for CPU1 and then for CPU2. It can be seen that the system with CPU1 has a linear increase of the aggregated throughput for offered loads below 225 Mbps. At this point the CPU utilization is 100% while the bus utilization is around 50% and the systems enters into a saturation state. If we further increase the offered load the throughput decreases until a live lock condition appears, at an offered load of 810 Mbps. During the saturation state most losses occur in the IP input buffer. The system with CPU2 gets its bus saturated before its CPU at an offered load of 500 Mbps. The system behavior for increasing offered loads depends on which priorities are used by the bus arbiter. Summarily, the basic system cannot provide a fair share of the resources when it is in saturation. Fig.5.b shows results for the system with a WFQ scheduling for the CPU and the BUG mechanism for controlling bus usage. We see that the obtained results correspond to almost an ideal behavior, as under saturation throughput does not decrease with increasing offered loads and the system achieves a fair share of both router resources: CPU and bus.

5 Conclusions

Under quite normal operation conditions for today's PC hardware and telecommunication links, the plain PCI bus arbitration mechanism impedes a software router to fulfill QoS guarantees. The mechanism that we proposed and called BUG, for bus usage control, is effective in controlling the bus share between different flows.

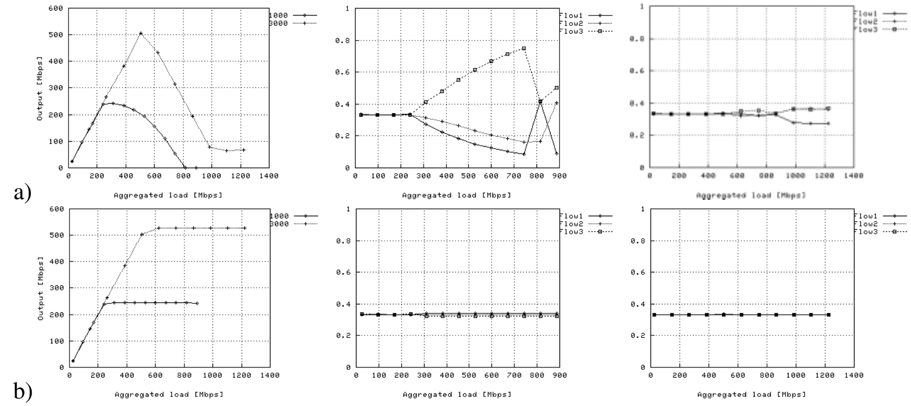


Fig. 2. Performance results for (a) base BSD router (b) a router with WFQ for the CPU and BUG for the I/O bus. The charts at the left contrast the router throughput when it uses a CPU of 1GHz and a CPU of 2GHz. The charts at the middle and at the right show the throughput share obtained by each of the three flows described in the main text. The charts at the middle are for a router using a 1GHz CPU, while the charts at the right are for a router using the 2GHz CPU.

When we use this mechanism in combination with the known techniques for CPU usage control, we obtain a nearly ideal behavior of the share of the software router resources for a broad range of workloads.

References

1. Indiresan, A. Mehra and K. G. Shin, “Receive Livelock Elimination via Intelligent Interface Backoff”, December 1997, http://citeseer.nj.nec.com/366416.html
2. O. I. Lepe and J. García, “A Performance Model of a PC-based IP Software Router”, to appear at Proc. IEEE ICC2002, April 2002.
3. M. L. Loeb, A. J. Rindos, W. G. Holland and S. P. Woolet, “Gigabit Ethernet PCI Adapter Performance”, IEEE Network, 15(2): 42-47, March/April 2001.
4. Mogul and K. K. Ramakrishnan, “Eliminating receive livelock in an interrupt-driven kernel”, ACM Trans. Computer Systems, 15(3): 217-252, August 1997.
5. K. Parekh and R. G. Gallager, “A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks- The Multiple Node Case”, Proc. IEEE INFOCOM 1993, pp. 521-530 vol.2
6. X. Qie, A. Bavier, L. Peterson and S. Karlin, “Schedulling Computations on a Software-Based Router”, Proc. SIGMETRICS 2001, June 2001.

Multiple Access in Ad-Hoc Wireless LANs with Noncooperative Stations

Jerzy Konorski

Technical University of Gdansk
ul. Narutowicza 11/12, 80-952 Gdansk, Poland
jekon@pg.gda.pl

Abstract. A class of contention-type MAC protocols (e.g., CSMA/CA) relies on random deferment of packet transmission, and subsumes a deferment selection strategy and a scheduling policy that determines the winner of each contention cycle. This paper examines contention-type protocols in a noncooperative an ad-hoc wireless LAN setting, where a number of stations self-optimise their strategies to obtain a more-than-fair bandwidth share. Two scheduling policies, called RT/ECD and RT/ECD-1s, are evaluated via simulation It is concluded that a well-designed scheduling policy should invoke a noncooperative game whose outcome, in terms of the resulting bandwidth distribution, is fair to non-self-optimising stations.

1 Introduction

Consider N stations contending for a wireless channel in order to transmit packets. In a cooperative MAC setting, all stations adhere to a common contention strategy, C, which optimises the overall channel bandwidth utilisation, U: $C=\mathrm{argmax}U(x)$. In a noncooperative MAC setting, each station i self-optimises its own bandwidth share, U_i: $C_i^*= \mathrm{argmax}U_i(C_1^*,...,C_{i-1}^*,x,C_{i+1}^*,...,C_N^*)$. C_i^* is a *greedy* contention strategy and $(C_1^*,...,C_N^*)$ is a *Nash equilibrium* [3] i.e., an operating point from which no station has incentives to deviate unilaterally. Note that noncooperative behaviour thus described is *rational* in that a station intends to improve its own bandwidth share rather than damage the other stations'. This may result in unfair bandwidth shares for stations using C. For other noncooperative wireless settings, see [1,4].

In ad-hoc wireless LANs with a high degree of user anonymity (for security reasons or to minimise the administration overhead), noncooperative behaviour should be coped with by appropriate contention protocols. A suitable communication model is introduced in Sec. 2. The considered contention protocol under the name Random Token with Extraneous Collision Detection (RT/ECD) involves voluntary deferment of packet transmission. We point to the logical separation of a deferment selection strategy and a scheduling policy that determines the winner in a contention cycle. Sec.

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 1141-1146, 2002.

3 outlines a framework for a noncooperative MAC setting. A scheduling policy called RT/ECD-1s is described in Sec. 4 and evaluated against the RT/ECD policy in terms of the bandwidth share guaranteed for a cooperative (c-) station (using C) in the presence of noncooperative (nc-) stations (using C_i*). Sec. 5 concludes the paper.

2 Noncooperative MAC Setting with RT/ECD

Our 'free-for-all' communication model consists of the following non-assumptions:

- neither N nor stations' identities need to be known or fixed,
- except for detecting carrier, a station need not interpret any packet of which it is not an intended (uni- or multicast) recipient.

This allows for full encryption and/or arbitrary encoding and formatting among any group of stations. To simplify and restrict the model we assume in addition

- single-hop transfer of packets with full hearability, and
- a global slotted time axis.

Any station is thus able to distinguish between v- and c-slots sensed (for 'void' and 'carrier'). An intended recipient of a successful transmission recognises also an s-slot (for 'success') and reads its contents. This sort of binary feedback allows for *extraneous collision detection* in the wireless channel as employed by the following RT/ECD protocol (Fig. 1). In a protocol cycle, a station defers its packet transmission for a number of slots from the range 0..D−1, next transmits a 1-slot *pilot* and senses the channel in the following slot. On sensing an s-slot containing a pilot, any intended recipient transmits a 1-slot *reaction* (a burst of non-interpretable carrier), while refraining from reaction if a v- or c-slot is sensed. A reaction designates the sender of a successful pilot as the winner and prompts it to transmit its packet in subsequent slots; a v-slot will mark the termination of this protocol cycle. If pilots collide, no reaction follows and the protocol cycle terminates with a no-winners outcome. In a full-hearability environment, RT/ECD operates much like CSMA/CA in the IEEE 802.11 Distributed Coordination Function [2], with the pilot/reaction mechanism resembling the RTS/CTS option. Note, however, that it is to provide ACK functionality rather than solve the hidden terminal problem; moreover, pilots only need to be interpreted by intended recipients, while reactions are non-interpretable.

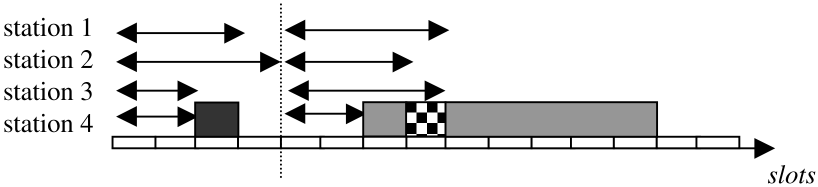


Fig. 1. RT/ECD, a no-winners protocol cycle followed by one where station 4 wins

To account for noncooperative behaviour, we assume that

- *NC* out of *N* stations are nc-stations that may use greedy deferment selection strategies (*NC* need not be known or fixed),
- the c-stations use a standard deferment selection strategy *S*, defined by the probabilities π_l of selecting a deferment of *l* slots ($l \in 0..D-1$), and
- all stations adhere to a common scheduling policy.

A simple greedy strategy might consist in introducing a downward $bias \in 0..D-1$ to the deferment distribution e.g., $\pi_0' = \pi_0 + ... + \pi_{bias}$ and $\pi_l' = \pi_{l+bias}$ for $l>0$. As shown in Sec. 4, this may leave the c-stations with a tiny fraction of the bandwidth share they would obtain in a cooperative setting (with $NC=0$).

3 Framework for a Noncooperative MAC Setting

Besides pursuing a greedy deferment selection strategy, an nc-station might try various 'profitable' departures from the protocol specification – for example, pretend to have transmitted a pilot and sensed a reaction. In RT/ECD-like protocols, however, such cheating must involve making false claims as to the presence or absence of carrier on the channel, which is easily verifiable. Therefore it suffices to design a scheduling policy so as to minimise the benefits of any conceivable greedy strategy vis-a-vis *S*. A greedy strategy can be expected to be

- *isolated* i.e., not relying on collusion with other nc-stations, and
- *rational*, meaning that deferment selection rules observed to increase own bandwidth share are more likely to be applied in the future, however, to stay responsive to a variable environment, no rules are entirely abandoned [3].

A reasonable scheduling policy is constrained to be

- *nontrivial*, in that no deferments should be known a priori to render other deferments non-winning (note that RT/ECD is a counterexample, deferment of length 0 being 'fail-safe'), and
- *incentive compatible*, in that channel feedback up to any moment in the deferment phase should not discourage further pilots (as a counterexample, imagine a scheduling policy whereby a second-shortest deferment wins).

Let $U_c(NC)$ be the bandwidth share obtained by a generic c-station in the presence of *NC* nc-stations. A fair and efficient scheduling policy is one that ensures $U_c(NC)$ '≥' $U_c(0)$ '≥' $U_c(0)|_{RT/ECD}$ for any *NC* and any greedy strategy, where '≥' reads 'not less or at least tolerably less than.' This means that the presence of nc-stations should not decrease a generic c-station's bandwidth share by an amount that its user would not tolerate. The latter 'inequality' implies that protection against nc-stations should not cost too much bandwidth in a cooperative setting, RT/ECD being a reference policy supposed, by analogy with IEEE 802.11, to perform well in a cooperative setting.

4 Evaluation of the RT/ECD-1s Scheduling Policy

While RT/ECD prevents any station from winning if a collision of pilots occurs, in RT/ECD-1s the first successful pilot wins no matter how many collisions precede it. A protocol cycle is illustrated in Fig. 2. A slot occupied by a pilot (or a collision of pilots) is paired with a following one, reserved for reactions. Stations whose pilots were not reacted to back off until the next protocol cycle. The lack of a second chance to transmit a pilot in the same protocol cycle creates a desirable 'conflict of interest' for an nc-station selecting its deferment. RT/ECD-1s is arguably nontrivial and incentive compatible. (A family of similar policies can be devised whereby the n^{th} successful pilot wins, or the last one if there are less than n; of these, RT/ECD-1s yields the best winner outcome vs. scheduling penalty tradeoff.)

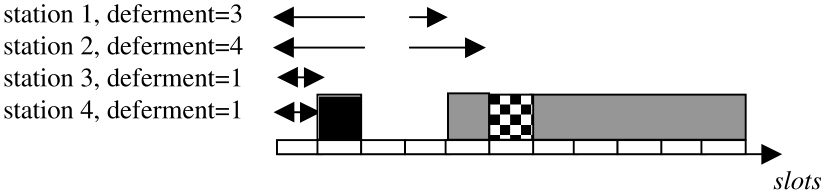


Fig. 2. RT/ECD-1S protocol cycle: stations 3, 4 back off when no reaction follows; station 1's first successful pilot wins (deferments are frozen during reaction slots)

In a series of simulation experiments, simple models of c- and nc-stations were executed to evaluate RT/ECD-1s against the backdrop of RT/ECD. In each simulation run, D=12, N=10 and $NC \in 0..N-1$ were fixed and packet size was 50 slots. Symmetric heavy traffic load was applied with one packet arrival per station per protocol cycle. The strategy S at the c-stations used a truncated geometric probability distribution over $0..D-1$ i.e., $\pi_l = const. \cdot q^l$ with parameter q=0.5, 1 or 2 (referred to symbolically as 'aggressive,' 'moderate' and 'gentle'). Two isolated and rational greedy strategies were experimented at nc-stations: Biased Randomiser (BR) and Responsive Learner (RL). The former introduced a downward bias as explained in Sec. 2; the *bias* value was optimised on the fly at each nc-station and occasionally wandered off the optimum to keep the strategy responsive to possible changes in other stations' strategies. The latter mimicked so-called *fictitious play* [5] by selecting deferments at random based on their winning chances against recently observed other stations' deferments. Once selected, a deferment was repeated consistently throughout the next update period of UP=20 protocol cycles. For simplicity, strategies were configured uniformly within all stations of either status, producing two noncooperative game scenarios: S vs. BR and S vs. RL.

Ideally, $U_c(NC) \equiv 1/N$ of the channel bandwidth. Scheduling penalties cause this figure to drop even in a cooperative MAC setting (at NC=0), whereas nc-stations may bring about a further decrease. For the S vs. BR scenario, Fig. 3 (*left*) plots $U_c(NC)$ (normalised with respect to $1/N$) as measured after the nc-stations have

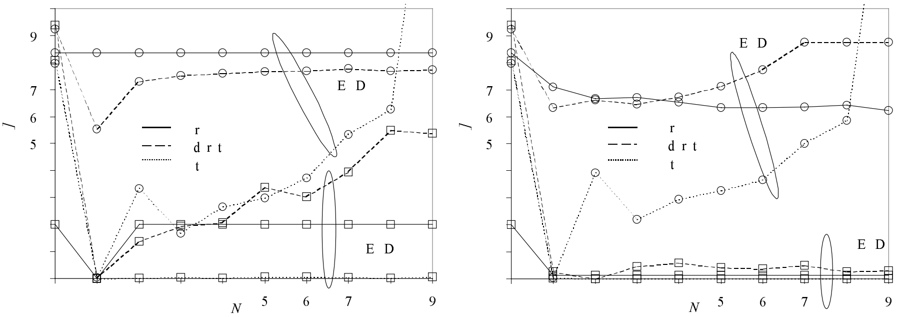


Fig. 3. C-station bandwidth share as a function of *NC*, *left*: S vs. BR, *right*: *S* vs. RL

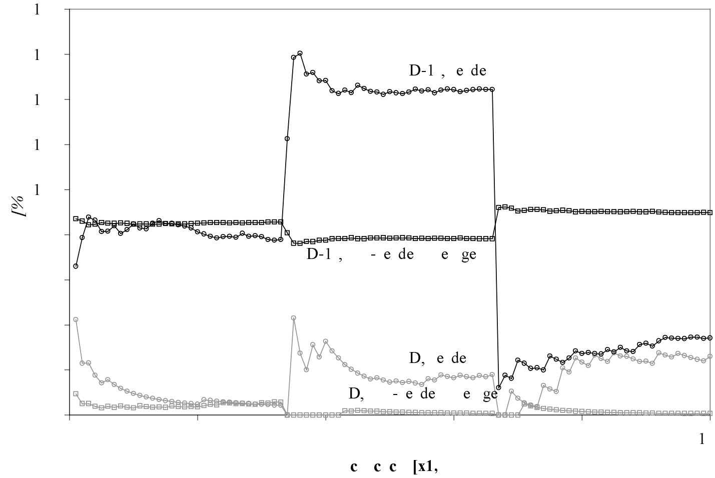


Fig. 4. RL vs. RL: Stackelberg ’leader’ scenario

reached a Nash equilibrium with respect to *bias*. Note that while RT/ECD-1s is generally superior to RT/ECD, much depends on the parameter *q*: the ’gentle’ value is not recommended, especially for a small *N*, while for the ’aggressive’ value, the nc-stations detect that the optimum *bias* is 0, hence $U_c(NC)$ remains constant. Also, RT/ECD-1s has difficulty coping with *NC*=1. Fig. 3 (*right*) presents similar results for the *S* vs. RL scenario. Observe that under RT/ECD-1s, nc-stations’ increased intelligence does not worsen $U_c(NC)$ significantly, which it does under RT/ECD. Again, much depends on *q*: although the ’moderate’ value pays off in a cooperative setting, the ’aggressive’ value offers more uniform guarantees for $U_c(NC)$ across various *N*. Lose-lose situations (with both the c- and nc-station bandwidth shares below $U_c(0)$) were observed under RT/ECD owing to this policy not being nontrivial.

Fig. 4 presents an RL vs. RL scenario where, after a third of the simulation run, one nc-station captures more bandwidth by lengthening its update period tenfold whenever a deferment of length 0 is selected. In doing so, it becomes a so-called Stackelberg 'leader' [5]. A form of protection, switched on after another third of the simulation run, is for a c-station to monitor its own and other stations' win counts over the last update period. If the former is zero and the latter nonzero, the station temporarily resorts to *S* with the 'aggressive' *q*. Under RT/ECD-1s, this quickly results in the 'leader' obtaining a less-than-fair bandwidth share. Under RT/ECD the protection is ineffective; moreover, the overall bandwidth utilisation remains poor.

5 Conclusion

Ad-hoc wireless LAN systems, with their preferences to user anonymity and a lack of tight administration, potentially constitute a noncooperative MAC setting. For a class of contention protocols relying on random deferment of packet transmission, c-stations are vulnerable to unfair treatment by nc-stations, which use greedy deferment selection strategies. The design of a scheduling policy has been shown to be quite sensitive in this respect. A framework for a reasonable scheduling policy and greedy strategies that might be expected from nc-stations has been outlined. A slotted-time scheduling policy called RT/ECD, analogous to CSMA/CA with the RTS/CTS option, and an improved variant thereof called RT/ECD-1s have been evaluated under heavy load to find that the latter guarantees the c-stations a substantially higher bandwidth share. This it does assuming that nc-stations behave rationally and seek a Nash equilibrium. In the experiments, RT/ECD-1s coped well with nc-stations using a randomisation bias or a fictitious play-type greedy strategy.

Several directions can be suggested for future work in this area:

- a game-theoretic study of RT/ECD-like scheduling policies aimed at establishing the mathematical properties of the underlying noncooperative games,
- model extensions to include multihop wireless LAN topologies (in particular, dealing with the problem of hidden stations); development of a suitable extension of RT/ECD-1s is under way, and
- access delay analysis to investigate the issues of QoS support.

References

1. Heikkinen, T.: On Learning and the Quality of Service in a Wireless Network. In: Proc. Networking 2000, Springer-Verlag LNCS 1815 (2000), 679-688
2. IEEE 802.11 Standard (1999)
3. Kalai, E., Lehrer, E.: Rational Learning Leads to Nash Equilibrium, Econometrica 61 (1993), 1019-1045
4. MacKenzie, A.B. and Wicker, S.B.: Game Theory and the Design of Self-Configuring, Adaptive Wireless Networks, IEEE Comm. Magazine, 39 (2001), 126-131
5. Milgrom, P., Roberts, J.: Adaptive and Sophisticated Learning in Normal Form Games, Games and Economic Behaviour 3 (1991), 82-100

Next Generation Networks and Services in Slovenia

Andrej Kos, Janez Bešter, and Peter Homan

University of Ljubljana, Faculty of Electrical Engineering, Laboratory of Telecommunications, Tržaška 25, 1000 Ljubljana, Slovenia,
{andrej.kos, janez.bester, peter.homan}@fe.uni-lj.si
http://www.ltfe.org

Abstract. This paper provides an overview of development of telecommunications in Slovenia. Major systems, networks and services are briefly considered. The combination of own generic research and critical mass of knowledge had and still has a very positive influence the on development of telecommunications in Slovenia. We propose a two-level network architecture consisting of a simplified data forwarding plane and service control plane. Future technological development and the proposed role of Slovenia as a regional telecommunications hub are presented.

1 Introduction

Recent years have been marked with significant advances in telecommunications. The main reasons for fast development are:

1. Fast development of new technologies
2. Rapidly falling prices of networking equipment and bandwidth
3. Rapidly falling prices of services
4. Changing the basic platform of telecommunications from connection oriented networks to connectionless, packet-based networks
5. Convergence
6. Rapid shift of importance from technology towards services
7. Deregulation and liberalization

In 2000 there were still some doubts about the general development path of telecommunications. Two scenarios were possible: evolution and revolution. Revolutionary scenario anticipated the advent of new, small, specialized, and technically very advanced actors. The services would all be provided over IP infrastructure. Evolutionary scenario anticipated gradual transformation of classical telecommunications in 10 to 15 years from PSTN/ISDN-centric to IP-centric companies. It is now clear that the future development in telecommunications will follow evolutionary path. Telecoms, on contrary to new players, typically have large investments in embedded base and strong revenue-generating existing services (voice) that help fund extensive and expensive network as well as service upgrades. Areas where investment is particularly intense are mobile, broadband, and Internet.

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 1147–1152, 2002.

2 State of Telecommunications in Slovenia

Slovenia has relatively well developed telecommunications sector. Some important characteristics of Slovenian telecommunications are summarized in Table 1.

Table 1. Main telecommunications indicators

Indicators	End of 2001
Population	1.971.000
GDP/inhabitant	€10.840
ISDN/PSTN density	45 %
Mobile density	70 %
Internet users	35 %
CaTV penetration (households)	37,5 %
Digitalization	99 %

Slovenia belongs among 15 countries in the world that have generic telecommunications development and are capable of developing, producing and exporting advanced telecommunications systems and solutions. There is tight cooperation between industry and academic institutions. Combination of own generic research and critical mass of knowledge had and still has very positive influences on development of telecommunications in Slovenia.

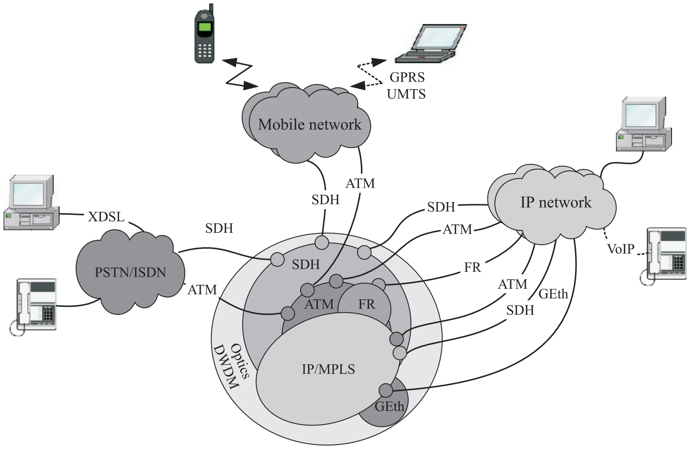


Fig. 1. Core infrastructure

Core infrastructure that supports all three main segments; fixed telephony, mobile, and IP is shown in Fig. 1. It is based on optical cable systems upgraded with different technologies on different layers, such as DWDM, SDH, FR, ATM, Gigabit Ethernet, MPLS, and IP. It is mainly provided by Telekom Slovenije. In lesser extent it is also provided by Elekto-Slovenija, Slovenian Railways, and Motorway Company in the Republic of Slovenia. The latter offer leased line services over SDH infrastructure. Fixed telephone network is currently still the most important Slovenian telecommunications infrastructure. At the end of 2000 digitalization rate reached 99 % and the PSTN/ISDN penetration is 45 %. The penetration of ISDN and centrex together is 7.3 %. Fixed telephone network is structured in two-level hierarchy; primary (PX) and secondary (SX), which is hierarchically higher than the primary. Broadband ADSL services over copper access network are available from the beginning of 2001.

Mobile communications are well developed with one of the highest penetration rates in Europe. Three mobile operators, Mobitel, Si.mobil, and Western Wireless International are operating in Slovenia. Service provider Debitel uses Mobitel's GSM network. At the end of 2001 the penetration rate of mobile users was over 70 %. Comparison of mobile penetration rates with other European countries is shown in Fig. 2. The data is valid for September 2001.

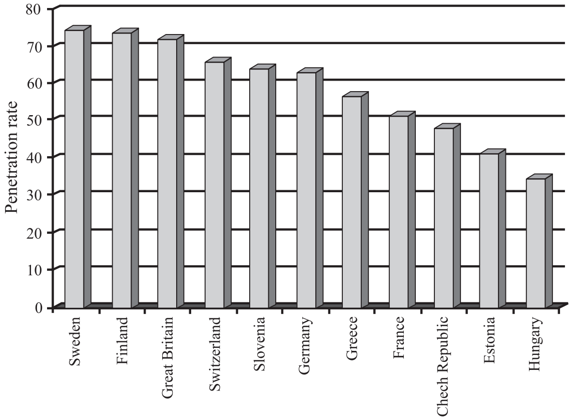


Fig. 2. Comparison of mobile penetration rates (September 2001)

As in the rest of the Europe, the number of people using the Internet continues to grow. In October 2001 there were some 700.000 (35 %) Internet users. A user for the above figure is defined as someone who has used the Internet at least once in the past three months. Of these 700.000 users,

- 500.000 use the Internet at least once per month
- 400.000 use the Internet at least once per week
- 300.000 use the internet on a daily basis

The biggest internet service provider in Slovenia is Siol. It manages the biggest core commercial network and currently offers dial-up access, leased lines, ADSL, and Ethernet access. In addition to different types of access to the Internet, Siol offers services such as VPNs, web hosting, all standard IP services, and many new application services, such as audio/video, e-commerce, and distance learning. The other big player in the field of Internet is Academic and Research Network of Slovenia. The main task of Arnes is development, operation and management of the communication and information network for education and research. There is a variety of smaller commercial ISPs that provide internet services, such as access to the Internet, web hosting, consulting and similar. Currently there are more than 100 CaTV operators in Slovenia, which provide services to around 250.000 Slovenian households and 750.000 users respectively. Thus the CaTV penetration rate is 37.5 %. In some urban areas the penetration rate is more than 90 %. However the great majority of operators are small companies owned by local communities.

3 Convergence

As shown in Fig. 1 telecommunications today are based on three pillars: fixed, mobile and IP. Technologically all three can support voice and data/internet. Up to now terminals for fixed telephone network were classical telephone terminals. With the advent of xDSL, the access telephone network is being used for broadband data as well. GSM mobile networks were primarily built to support voice, but with HSCSD, GPRS and UMTS more and more data traffic will be transported over mobile networks. In the past typical usage of IP networks was data, but with the advent of VoIP, IP networks are being used for voice as well. Especially it is expected that the boundary between mobile operators and Internet service providers will blur due to strong cross-area expansion. With the advent of ADSL there is also a similar blurring of the boundary between fixed operators and Internet Service providers.

General convergence trends that can be identified are:

1. Voice is migrating from fixed to mobile networks (overall voice is growing, whereas there is a decline in fixed voice)
2. Fixed networks will be used for broadband data
3. IP networks are converging into a common infrastructure for all existing and new services through implementation of MPLS

4 Future Development

Till the end of 2000 the core network was working mainly in a connection oriented transport fashion, Anticipated technological evolution of core network in general is presented in [1,2], where it should be noted that although today's vision of next generation core network is IP/MPLS/GMPLS over DWDM, existing, proven and well-known technologies such as SDH, ATM and Gigabit Ethernet would still be used for a long time. As discussed below only their role might be slightly different.

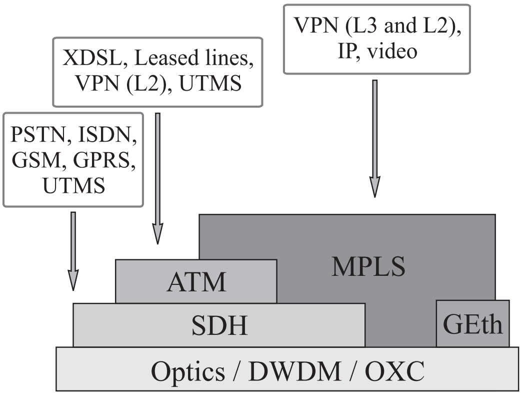

VPN (L3 and L2), IP, video
XDSL, Leased lines, VPN (L2), UTMS
PSTN, ISDN, GSM, GPRS, UTMS
MPLS
ATM
SDH
GEth
Optics / DWDM / OXC

Fig. 3. Concept of contemporary network architecture and its main usage

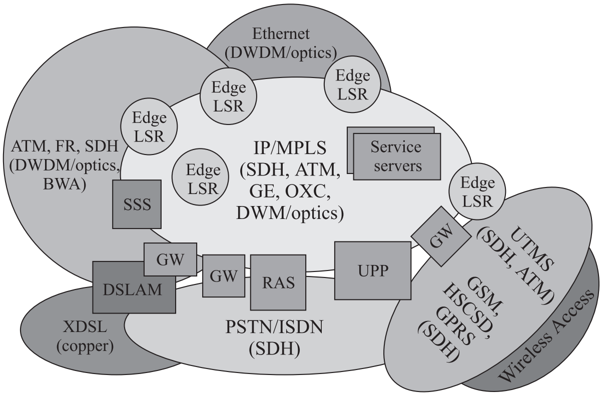


Fig. 4. Next generation network

According to general evolution of core network the concept of future network architecture and its main usage will change as shown in Fig. 3. The concept is based on the following facts:

1. IP protocol has become the convergence layer for majority of services
2. MPLS and its generalization GMPLS have become the core technology of choice that in addition to connection oriented approach support many new functionalities in terms of routing, signaling, control and QoS support
3. ATM as a layer 2 technology is with ADSL and ATM switches at customer sites migrating towards access
4. Voice services will still for some time be accessed via classical terminals, mainly mobile. VoIP functionality will be through media gateways first introduced mainly in the core as voice trunking

In [3] framework for next generation network is proposed. Logically it is a two-level network architecture, which consists of service control layer and transport layer. The transport is service independent. We propose next generation network, of which technology aware view is shown in Fig. 4 [4,5] (extended version of this paper can be found on http://www.ltfe.org/pdf/networking2002_extended.pdf). Most of the intelligence is in edge devices. Edge devices' functionalities include termination of different access technologies, data format adaptation for transport over core network, service gateways, such as QoS mappings, connection admission control, classification, metering, marking, dropping, authorization, accounting, fire-walling, address translation, security, and others.

5 Conclusion

In the article the overview of development in the field of telecommunications in Slovenia is presented. The combination of own generic research and critical mass of knowledge had and still has very positive influences on development of telecommunications. We propose a two-level network architecture consisting of a simplified data forwarding plane and service control plane. Service control plane is mostly implemented in edge devices, in the form of different gateways and servers. Slovenia with less than 2 million inhabitants is relatively small market and will in global markets have to find its place in niche segments. With a lot of technological know-how, unique geographic position, a lot experience, and good relationships with all neighboring countries, one among most important niche segments is being a telecommunications hub.

Acknowledgements. The authors would like to thank Matej Eljon and the anonymous reviewers for their thoughtful review and useful suggestions.

References

1. Kos, A., Bešter, J.: Role of MPLS in Modern Telecommunications Networks. International Symposium Viable Telecommunications VITEL 2000, Technologies and Communication for the Online Society, Ljubljana, Slovenia (2000) C45-C49
2. Banerjee, A., et al.: Generalized Multiprotocol Label Switching: An overview of Routing and Management Enhancements. IEEE Commun. Mag., vol. 39, no. 1 (2001) 144-150
3. Moridera, A., Murano, K., Mochida, Y.: The Network Paradigm of the 21^{st} Century and Its Key Technologies, IEEE Commun. Mag., vol. 38, no. 11 (2000) 94-98
4. Rockström, A.: Technology as a Driver for New Business Logic. IEEE Commun. Mag., vol. 38, no. 11, (2000), 100-104
5. Žurbi, R.: Signalling and Control Protocols in Next Generation Networks. M.Sc. Thesis, Faculty of Electrical Enginering, University of Ljubljana, Ljubljana, Slovenia (2001)

Minimizing the Routing Delay in Ad Hoc Networks through Route-Cache TTL Optimization

Ben Liang and Zygmunt J. Haas

School of Electrical and Computer Engineering, Cornell University, Ithaca, NY 14853, USA
{liang, haas}@ece.cornell.edu

Abstract. This paper addresses the issue of minimizing the routing delay in ad hoc on-demand routing protocols through optimizing the Time-to-Live (TTL) interval for route caching. An analytical framework is introduced to compute the expected routing delay when the source node has a cached route with a given TTL value. Furthermore, a numerical method is proposed to determine the optimal TTL of a newly discovered route cached by the source node. We present simulation results that support the validity of our analysis.

1 Introduction

Node mobility and the lack of topological stability make the routing protocols previously developed for wireline networks unsuitable for ad hoc networks[9][8][11]. A popular family of ad hoc routing protocols are the reactive routing protocols, also called *on-demand* routing protocols. In these protocols a node is not required to maintain a routing table (although route caches may be kept), but instead a route query process is initiated whenever it is needed. Routing protocols such as ABR, AODV, DSR, the IERP of ZRP, and TORA are examples of reactive protocols[11][1].

In an on-demand routing protocol, a newly discovered route should be cached, so that it may be reused the next time that the same route is requested. However, prolonged storage of a route cache may render it obsolete. When an invalid route cache is used, extra traffic overhead and routing delay is necessary to discover the broken links. Depending on the implementation details, data and/or control packets are delivered over part of the cached route that is still valid, before the broken link can be discovered.[2]

One approach to minimize the effect of invalid route cache is to purge the cache entry after some Time-to-Live (TTL) interval. If the TTL is set too small, valid routes are likely to be discarded, but if the TTL is set too large, invalid route-caches are likely to be used. Thus, an algorithm that optimizes the TTL setting is necessary for the optimal performance of an on-demand routing protocol.

As far as we are aware, there is very little reported work in literature that addresses the issue of ad hoc route-cache TTL optimization. Most existing on-demand protocols, such

[1] Due to the page limit, the individual references to these protocols are omitted.

[2] It is possible to employ *proactive* route-cache invalidation initiated by the up-stream node of a broken link, whether or not the link is part of an active route presently delivering data. However, this can lead to large control overhead when the network topology changes frequently. Proactive techniques are outside the scope of this paper.

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 1153–1158, 2002.

as AODV, DSR, and TORA, employ route caching in various forms. In AODV, a discovered route is associated with an "active route time-out" value that dictates the duration within which the route can be used. This time-out value is static and identical throughout the network. In DSR and TORA, a cached route is kept indefinitely (i.e. TTL=∞), until a broken link in the route is detected during data transmission. In this work, we study the TTL optimization adaptive to each cached route.

In [10], case study based on DSR has suggested that route caching can reduce the average latency of route discovery by more than 10-fold. Further simulation studies reported in [1], [7], [4], [2], and [3] have confirmed the effectiveness of route caching in on-demand routing protocols. However, [7], [4], [2], and [3] have also drawn the conclusion that the indefinite route-cache, as is employed in DSR, can lead to many stale routes and hence degrade the routing performance. In addition, [2] and [3] have demonstrated the need for determining a suitable time for the route-cache expiration. The simulation results in [3] have further shown a case study of the optimal route-cache expiry time obtained by exhaustive search. In this work, we approach the problem of adaptive route-cache TTL optimization through analytical studies.

We consider the problem of optimizing the TTL of a cached route in order to minimize the expected routing delay of the next request of the same route (i.e., the same source and destination pair). In Section 2, we explain the network model under consideration. In Section 3, we introduce analytical and numerical frameworks to compute the optimal TTL and the corresponding expected routing delay. In Section 4, we present simulation results that support the validity of our analysis, study the system parameters that affect the optimal TTL, and show the performance gain achieved by the optimal TTL. Finally, concluding remarks are provided in Section 5.

2 Network Model

We consider a mobile ad hoc network consisting of a set V of nodes. At any time instant, an edge (u, v), where $u, v \in V$, exists if and only if node u can successfully transmit to node v. In this case, we say that the link from node u to node v is *up*. Otherwise, the link is *down* or has *failed*.

In the modeling of general communication networks, it is usually assumed that all edge failures are statistically independent [6]. The modeling of dependent link failures generally requires an exponentially large number of conditional probability distributions. Therefore, though unrealistic, the independence assumption greatly simplifies the analysis of network performance. In this paper, we assume that all links have independent and identical up-time distribution $F_u(t)$ and down-time distribution $F_d(t)$.

We assume that route requests to a destination node n_d arrive at the source node n_s as a stream that has identically distributed inter-arrival intervals with a general distribution $F_a(t)$. We consider only non-trivial networks where the average time between topology changes is smaller than the average route-search delay.[3] Therefore, we assume that the route-request inter-arrival time is much larger than the route-search delay, since, otherwise, a valid route is already found at the last route request. Namely, a burst of

[3] Otherwise, the only suitable routing approach is to flood data packets throughout the network.

data packet train sent to a common destination within a very small time frame would constitute a single route request.

When a route request is made due to a data packet arrival, if n_s has a cached route to n_d, it immediately sends out the data packet using the cached route. If the cached route is valid, we assume that this operation does not incur any routing delay. However, if the cached route is invalid, the intermediate node on the up-stream end of a failed link notifies n_s via a route-error packet. In this case, and in the case that n_s does not have a cached route to n_d, the pre-defined routing protocol [4] is employed to search for a new route to n_d. We further assume that n_s renews or re-computes the TTL of a cached route to n_d each time a packet is successfully sent through the cached route. A cached route is purged when its TTL expires.[5]

We assume that all data and control packet transmissions across a link incur an average delay of L seconds.[6]

3 Optimizing the Route-Cache TTL to Minimize Routing Delay[7]

3.1 Computing the Expected Routing Delay

Suppose the source node n_s has a cached route to the destination node n_d, which is validated by the last route request and has a TTL value of T seconds. Let D be the number of hops in this route.

Let the next route request to n_d arrive at time t_a after the n_s-to-n_d route is cached. Then, from Section 2, t_a has distribution $F_a(t)$. Let $f_a(t)$ be the density function of t_a, and let ${f_a}^*(s)$ be the Laplace transform of $f_a(t)$. Furthermore, let $f_c(t)$ be the density function of t_a given $t_a < T$. Then, the Laplace transform of $f_c(t)$ is ${f_c}^*(s) = -\frac{1}{F_a(T)} \sum_{\xi \in \text{poles of } {f_a}^*(s-z)} Res_{z=\xi} \frac{1-e^{-zT}}{z} {f_a}^*(s-z)$, where $Res_{z=\xi}$ denotes the residue at the pole $z = \xi$.

Let $f_u(t) = dF_u(t)/dt$ be the density function of the link up-time and ${f_u}^*(s)$ be its Laplace transform. The residual lifetime of a link in the cached route has the density function $f_r(t) = \frac{1}{\mu_u}[1 - F_u(t)]$, where μ_u is the mean up-time of a link. The Laplace transform of $R_r(t) = 1 - \int_0^t f_r(\tau)d\tau$ is ${R_r}^*(s) = \frac{1}{s} - \frac{1}{\mu_u s^2}[1 - {f_u}^*(s)]$.

Let X_i be the minimum of the residual lifetimes of the first i links in the cached route. Let $f_{X_i}(t)$ and $F_{X_i}(t)$ be its density and distribution functions, and let ${f_{X_i}}^*(s)$

[4] The exact mechanism of the on-demand routing protocol is not important here.

[5] Some on-demand protocols allow an intermediate node that has a cached route to the destination reply to the source initiated route request. Such schemes have been shown to significantly improve the routing performance. However, the quantitative effect of the stale routes provided by the intermediate nodes is not well understood. Therefore, in this work, we only consider the TTL of route caches kept by a source node.

[6] For example, in the case study of [10], the average delay is shown to be 14.5 ms/hop. Although packets may be different in length, in a wireless ad hoc network operating at medium to high load, the predominant factor in the aggregate delay of packet transmission across a link is the queuing delay in the MAC layer due to the contention of the shared wireless medium.

[7] Due to the page limit, in this section, we only give a brief outline of the important results and leave out the details to the long version of this paper.

and $F_{X_i}{}^*(s)$ be the corresponding Laplace transforms, respectively. Let $R_{X_i}(t) = 1 - F_{X_i}(t)$ and $R_r(t) = 1 - F_r(t)$, and $R_{X_i}{}^*(s)$ and $R_r{}^*(s)$ be their Laplace transforms, respectively. Then $f_{X_i}{}^*(s)$ can be determined through the following recursion: $R_{X_i}{}^*(s) = -\sum_{\xi \in \text{poles of } R_r{}^*(s-z)} Res_{z=\xi}\, R_{X_{i-1}}{}^*(z) R_r{}^*(s-z)$, along with $f_{X_i}{}^*(s) = sF_{X_i}{}^*(s) - F_{X_i}{}^*(0+) = 1 - sR_{X_i}{}^*(s)$.

Let $Q_i(T)$ be the probability that, when a route request arrives before the TTL expires, the first i links of the cached route have not failed. We can obtain $Q_i(T) = -\sum_{\xi \in \text{poles of } f_{X_i}{}^*(-s)} Res_{s=\xi}\, \frac{f_c{}^*(s)}{s} f_{X_i}{}^*(-s)$. Define $Q_0(T) = 1$. The expected routing delay of the next route request, when the TTL of a D-hop cached route is set to T, is $C(T) = 2L\left[D + F_a(T)\sum_{i=1}^{D} i\left(Q_{i-1}(T) - Q_i(T)\right) - F_a(T)Q_D(T)D\right]$.

The above analytical framework provides a means for evaluating the expected routing delay given the TTL value. However, it is very likely that the optimal TTL value is more important to a system designer. In the next section, we provide a numerical method to compute the optimal TTL.

3.2 Determining the Optimal Route-Cache TTL

Let $q(\tau)$ be the probability that a given link in the cached route is still up at time τ after the last route request. The expected routing delay as defined in the previous section has the following alternate form: $C(T) = 2LD - 2L\int_0^T \left[2Dq^D(\tau) - \frac{q^D(\tau)-1}{q(\tau)-1}\right] f_a(\tau)d\tau$.

Since $q(\tau)$ is a decreasing function of τ and $0 \le q(\tau) < 1$, it is easy to verify that $C(T)$ is a convex function of T. Therefore, if we let $g[q(T)] = -\frac{1}{2Lf_a(T)}\frac{dC(T)}{dT} = 2Dq^D(T) - \frac{q^D(T)-1}{q(T)-1}$, the minimum of $C(T)$ is achieved when $g[q(T)] = 0$. Therefore, the optimal value of $q(T)$ is the root in $[0,1)$ of a function of the form $g(x) = 2Dx^D - \frac{x^D-1}{x-1}$. Given any value of D, a numerical method such as bisection or the Newton's method can be used to find this root. Since $q(T) = 1 - F_r(T)$, once the optimal value of $q(T)$ is determined numerically, the optimal TTL value can be found by reversing the density function of the residual lifetime of a link.

The above illustrates an important property of the optimal TTL: it does not depend on $f_a(t)$. This property significantly reduces the computational requirement of the adaptive, real-time route-cache TTL optimization performed by individual nodes in an ad hoc network.

4 Simulation and Numerical Evaluation

4.1 Simulation Model and Output Analysis

A simulation model is developed to validate the analytical model. It represents the link establishments and breakages in an ad hoc network based on the network model described in Section 2. In particular, we present the simulation results for a 300-node network where the link up and down-times between any pair of nodes are exponentially distributed with mean values $\mu_u = 1$ and $\mu_d = 48.8$ (i.e., the average node degree is 6). Given a source node, the destination node is chosen randomly with uniform distribution

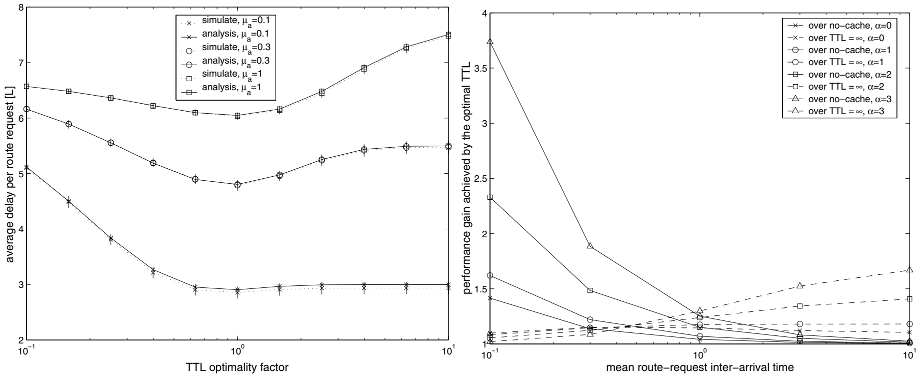


Fig. 1. (a) Expected routing delay (normalized to L) vs. the TTL optimality factor γ. The vertical lines represent the 99.95% confidence intervals. **(b)** Performance gain achieved by using the optimal route-cache TTL over TTL=0 and TTL=∞.

among all other nodes in the network. For a chosen source and destination node pair, the route-request inter-arrival time has distribution $f_a(t) = \frac{1}{\mu_a} e^{-\frac{t}{\mu_a}}$. We further define a TTL optimality factor γ, such that, when a new route is cached, its TTL is set to γT_{opt}, where T_{opt} is the optimal TTL value found as described in Section 3.2. The comparison between our analytical and simulation results is illustrated in Fig. 1(a).

Figure 1(a) validates both the analytical and the simulation models. In particular, the simulation results demonstrate that the minimal routing delay is indeed achieved at $\gamma = 1$, as expected from the analysis. The computed average delay per route request in some cases is 2% higher than the corresponding simulation outcome. This is due to the pessimistic assumption in the analytical model that once a link in a cached route fails, it does not become up again at the time of the next route request.

Figure 1(a) also suggests that the optimal TTL determination is the most important when the route-request inter-arrival time is moderate compared with the mean link-failure time. For systems with different parameter values, the results are similar to Fig. 1(a) and are omitted.

4.2 Performance Gain of the Optimal TTL

Using the proposed analytical framework, we can quantitatively study the advantage of optimizing the route-cache TTL. Due to the page limit, we are unable to show all results. In Fig. 1(b), we illustrate the performance gain of using the optimal TTL over the no route-cache system (TTL=0) and the never-expiring route-cache system (TTL=∞)[8], for different values of μ_a[9] and various traffic locality. In describing the traffic locality, we have used a power law distribution as follows. Let π_D be the probability that a given

[8] We define the performance gain as the ratio between the expected delay of using a non-optimal TTL and the expected delay of using the optimal TTL.

[9] We have scaled time such that $\mu_u = 1$. Therefore, $1/\mu_a$ represents the relative frequency of the route requests to the frequency of topology variation. Also note that the analytical results are valid for any μ_d as long as $\mu_d >> \mu_u$.

route request is made to a destination of D hops away. If D is upper-bounded by D_{max}, the probability distribution function of D is defined as $\pi_D = \frac{D^{-\alpha}}{\sum_{i=1}^{D_{max}} i^{-\alpha}}$, where a larger value of α indicates a higher level of locality. In this example, $D_{max} = 20$.

Figure 1(b) demonstrates that the performance gain is a fast increasing function of α. As a point of reference, when $\alpha = 3$ and $\mu_a = 1$, using the optimal TTL can reduce the routing delay of either a non-caching system or a never-expiring caching system by approximately 25%. Therefore, route-cache optimization is especially important in the design of *scalable* on-demand routing protocols for large mobile ad hoc networks, where it has been proven that the traffic pattern must be localized[5].

5 Conclusions

We have presented analytical and numerical methods to determine the expected routing delay and the optimal route-cache TTL for on-demand routing. The analysis is based on a random-graph model of mobile ad hoc networks. Our analytical results agree very well with the simulation results.

Through the proposed analytical framework, one can study the routing delay of a network given various system parameters. The results of our analysis have shown that the optimal route-cache TTL does not depend on the route-request frequency or inter-arrival distribution. Furthermore, our numerical results have demonstrated that optimizing the route-cache TTL is the most effective when the traffic pattern is localized.

References

1. J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva, “A performance comparison of multi-hop wireless ad hoc network routing protocols,” *ACM/IEEE MOBICOM*, 1998.
2. S. R. Das, C. E. Perkins, and E. M. Royer, “Performance comparison of two on-demand routing protocols for ad hoc networks,” *IEEE INFOCOM*, 2000.
3. Y.-C. Hu and D. B. Johnson, “Caching strategies in on-demand routing protocols for wireless ad hoc networks,” *ACM/IEEE MOBICOM*, 2000.
4. G. Holland and N. Vaidya, “Analysis of TCP performance over mobile ad hoc networks,” *ACM/IEEE MOBICOM*, August, 1999.
5. P. Gupta and P. R. Kumar, “The capacity of wireless networks,” *IEEE Trans. Information Theory*, vol. 46, no. 2, March 2000.
6. J. J. Kelleher, “Tactical communications network modeling and reliability analysis: overview,” JSLAI Report JC-2091-GT-F3, November 1991.
7. P. Johansson, T. Larsson, N. Hedman, B. Mielczarek, and M. Degermark, “Scenario-based performance analysis of routing protocols for mobile ad-hoc networks,” *ACM/IEEE Mobicom*, August 1999.
8. J. Jubin and J. D. Tornow, “ The DARPA packet radio network protocols,” *Proceedings of IEEE (Special Issue on Packet Radio Networks)*, vol. 75, pp. 21-32, January 1987.
9. B. M. Leiner, D. L. Nielson, and F. A. Tobagi, “Issues in packet radio network design,” *Proceedings of the IEEE*, vol. 75, pp. 6-20, January 1987.
10. D. A. Maltz, J. Broch, J. Jetcheva, and D. B. Johnson, “The effect of on-demand behavior in routing protocols for multihop wireless ad hoc networks,” *IEEE JSAC - Special Issue on Wireless Ad Hoc Networks*, vol. 17, no. 8, pp. 1439-1453, August 1999.
11. C. E. Perkins, ed., *Ad Hoc Networking*, Addison-Wesley Longman, 2001.

Long-Range Dependence of Internet Traffic Aggregates

Solange Lima, Magda Silva, Paulo Carvalho, Alexandre Santos, and Vasco Freitas

Universidade do Minho, Departamento de Informatica,
4710-059 Braga, Portugal
{solange, paulo, alex, vf}@uminho.pt

Abstract. This paper studies and discusses the presence of LRD in network traffic after classifying flows into traffic aggregates. Following DiffServ architecture principles, generic QoS application requirements and the transport protocol in use, a classification criterion of Internet traffic is established. Using fractal theory, the resulting traffic classes are analysed. The Hurst parameter is estimated and used as a measure of traffic burstiness and LRD in each traffic class. The traffic volume per class and per interface is also measured. The study uses real traffic traces collected at a University of Minho major backbone router in different periods of network activity.

1 Introduction

The diversity of quality of service (QoS) requirements of the actual and emergent services will force the network to differentiate traffic so that an adequate QoS level is offered. One of the most promising solutions proposed by the Internet Engineering Task Force (IETF) is the Differentiated Services architecture (DiffServ) [1], which aggregates traffic in a limited number of classes of service according to QoS objectives. This new network traffic paradigm poses renewed interest and challenge to network traffic analysis and characterisation. Although, several other studies focus on general Internet traffic characterisation, the effects of aggregating traffic in classes are still unclear. Will a particular traffic class be responsible for the behaviour expressed in [2]? Does aggregation affect burstiness at network nodes and links? The major objective of our work is to study fractal properties such as the long-range dependence (LRD) in Internet traffic aggregates.

Netflow[3] traffic samples collected at different time periods of network activity in a backbone router at the University of Minho were used. After establishing a traffic classification criterion based on a DiffServ model multi-field approach, all the samples are analysed applying that criterion. The time characteristics of each traffic class are studied resorting to the Mathematica software.

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 1159–1164, 2002.

2 The DiffServ Model

In the DiffServ model, network traffic is classified and marked using the DS-field [11]. This identifier determines the treatment or Per-Hop-Behaviour (PHB) [4,5] traffic will receive in each network node. The IETF has proposed the Expedited Forwarding PHB [4] and the Assured Forwarding PHB Group [5] (EF and AF PHBs), besides best-effort (BE PHB). The EF PHB can be used to build services requiring low loss, reduced delay and jitter, and assured bandwidth. The AF PHB group, consisting of four classes, can be used to build services with minimum assured bandwidth and different tolerance levels to delay and loss.

3 Network Traffic Characterisation

The knowledge of the network traffic characteristics as a whole and, in particular, of traffic aggregates is relevant to allow a proper network resources allocation and management, to help traffic engineering, traffic and congestion control, and to specify services realistically. In our study, the analysis is based on the fractal time series theory since recent studies related to the characterisation and modelling of network traffic point to the presence of self-similarity and LRD. This last property may directly affect the items highlighted above, with strong impact on queuing and on the nature of congestion [6].

3.1 Fractal Traffic Properties

Self-similarity expresses the invariance of a data structure independently of the scale that data is analysed. From a network traffic perspective, self-similarity expresses a new notion of burstiness, i.e. there is no natural length for a burst and bursty structure of traffic is maintained over several time scales.

As an example of processes which exhibit self-similarity and LRD, one may consider $X(t)$, an **asymptotically second order self-similar** stochastic process, with Hurst parameter $\frac{1}{2} < H < 1$, i.e., $\lim_{m\to\infty} \gamma(k) = ((k+1)^{2H} - 2k^{2H} + (k-1)^{2H})\frac{\sigma^2}{2}$. $X(t)$ has the following properties: *long-range dependence* - the autocorrelation function $\rho(k)$ decays hyperbolically ($\rho(k)$ is non-summable) $\lim_{k\to\infty} \frac{\rho(k)}{ck^{-\beta}} = 1$; *slowly decaying variances* - the variance of the aggregated series processes $X^{(m)}$, $X_k^{(m)} = \frac{1}{m}\sum_{i=km-m+1}^{km} X_i (k = 1, 2, ...$ and $m = 1, 2, ...)$, is expressed by $var(X^{(m)}) \sim var(X)m^{-\beta}$, with $c > 0$ constant, $\beta = 2 - 2H$, $0 < \beta < 1$.

H is commonly used to measure LRD, and a valuable indicator of traffic burstiness (burstiness increases with H). If $\frac{1}{2} <H< 1$ then an infinite persistence (indicating LRD) can be noticed. If $0 <H< \frac{1}{2}$ then no-persistent behaviour occurs, whereas if $H= \frac{1}{2}$ the variables are independent.

There are several methods to estimate the H parameter [7,2]. While methods such as the test of variances, the R/S statistic or the periodogram are based on graph analysis, the Whittle's estimator provides an analytical method to estimate H. Although the limitation of working with finite data samples and the error probability associated with graph based methods, they are widely used. The test of variances, which was used in this study, is based on the slowly decaying variance property, and H is obtained by $H = 1 - \frac{\beta}{2}$ resorting to a log-log plot of $(var(X^m), var(X)m^{-\beta})$.

3.2 Collecting and Preparing Traffic Samples

Traffic samples were collected from a major backbone router located at the Department of Informatics in University of Minho, using Cisco NetFlow tool [3]. NetFlow considers a flow as a unidirectional stream of packets from a source to a destination and records in each entry timing information, fields such as the source and destination IP addresses, port numbers, the protocol identifier, the input and output interfaces, and the number of packets and bytes sent.

The collection of traffic was carried out along different time periods and several days. These time periods were chosen reflecting typical levels of network activity (low: from 2 a.m. to 3 a.m.; medium: from 1 to 2 p.m. and from 10 to 11 p.m.; high: from 10 to 11 a.m. and from 3 to 4 p.m.). Each one-hour traffic sample is filtered by output interface according to the classification criterion presented in section 3.3 for different time intervals (100ms, 500ms, 1s and 10s). This process resulted in around 150 sample sets for analysis.

3.3 Traffic Classification Criterion

Due to economical and technical reasons the definition of a traffic classification criterion is a subjective task. For instance, for identical traffic types, a client may be willing to pay more than other to obtain a better service quality. Moreover, when a criterion is based on TCP/UDP/IP packet headers both packet fragmentation[1], packet encryption and the use of negotiated or unregistered application ports difficult classification[2]. Therefore, a classification criterion should be generic enough to be easily adopted and implemented. Most of the criteria suggest distinct classes for UDP and TCP traffic so that non-reactive and reactive applications do not compete for the same resources. Some go further suggesting that the duration of flows, the transmission rate and packet size characteristics should also be considered [9]. A classification method based on QoS application requirements such as delay or loss sensitivity is also proposed [10].

Considering the aspects above and the Type of Service (ToS) proposed for classical applications [10], our classification criterion is oriented to traffic aggregation which can easily be mapped to a class-based QoS architecture. As a first

[1] While fragmentation of UDP traffic is increasing, TCP traffic (around 85% of Internet traffic [8]) is virtually not fragmented due to the widespread use of MTU path discovery techniques and relatively small default packet sizes.

[2] The use of a modified IP Encapsulating Security Payload (which leaves protocol ports unchanged), the analysis of traffic at the control channels (which uses well-known ports) and/or the applications' usual range of ports or addresses might be possible solutions.

approach, the classification process distinguishes TCP from UDP traffic, and then, the generic applications requirements are taken into account. A filtering process based on more detailed rules to differentiate specific or proprietary applications (e.g. NetMeeting, Cisco IP/TV, and many other unicast or multicast applications) was left for further study. The resulting traffic classes are:

- **Class 1** - delay sensitive TCP traffic, resulting from interactive protocol applications such as Telnet, SSH or FTP control;
- **Class 2** - loss and throughput sensitive TCP traffic, resulting from bulk transfer protocol applications such as FTP data, DNS zone transfers, SMTP, POP, IMAP, NNTP;
- **Class 3** - essentially, HTTP traffic. Other TCP traffic not included in classes 1, 2 and 4 (a reduced volume) is mapped to this class;
- **Class 4** - priority traffic e.g. routing or management protocols (TCP/UDP);
- **Class 5** - generic UDP traffic e.g. TFTP, DNS, POP, IMAP or HTTP/UDP;
- **Class 6** - traffic from applications using transient ports (not allowing their classification) or UDP ports not covered by classes 4 and 5[3].

There is not a direct mapping between the defined classes and the DiffServ PHBs. Such mapping would depend on the administrative and contractual service policies. However, a possible match could be: classes 1 and 3 supported by high priority AF PHBs; class 4 by EF PHB; classes 2 and 5 by BE or low priority AF PHBs; class 6, could be either EF or AF depending on the relevance given to the diversity of applications.

4 Statistical Data Analysis

4.1 Traffic Volumes

Table 1 presents the percentage of traffic each class contributes for the total load in the router. The results show that the only class which contents is not clearly identified (Class 6) represents a small amount of traffic[4], which shows the broadness of the classification criterion proposed. As expected, classes 2 and 3, including mainly bulk and web traffic respectively, contribute heavily to the global router load. The results also show a correlation between the percentages of packets and bytes for all classes, although, classes 2 and 5 denote the presence of large and small packet sizes, respectively.

[3] According to [8], traffic from Real Audio and online game applications is likely the most significant one can consider in this class. Other real-time unicast/multicast traffic is also included here.

[4] For this reason, the statistical analysis of this class will be considered in a later stage of our study.

Table 1. Traffic volume per class

class	1	2	3	4	5	6
bytes	2.31 %	26.40 %	67.04 %	0.01 %	1.67 %	2.67 %
packets	2.37 %	18.95 %	68.41 %	0.04 %	5.94 %	4.27 %

4.2 Testing Long-Range Dependence

In order to analyse statistically whether a particular traffic class exhibits LRD tests of variance and autocorrelation analysis were carried out. For most of the samples, these tests illustrate similar results both for the analysis of the time series of packets and bytes.

Table 2. Percentage of samples and traffic volume with: a) $H < 0.45$, b) $0.45 \leq H \leq 0.5$, c) $0.5 < H < 0.7$, d) $H \geq 0.7$.

	Perc. of Samples					Traffic Volume			
classes	a)	b)	c)	d)		a)	b)	c)	d)
1	76.5%	0.0%	17.6%	5.9%		99.3%	0.0%	0.6%	0.1%
2	50.0%	6.7%	30.0%	13.3%		37.5%	3.3%	43.7%	15.5%
3	17.2%	6.9%	20.7%	55.2%		4.9%	6.6%	7.5%	81.0%
4	91.7%	8.3%	0.0%	0.0%		98.9%	1.2%	0%	0.0%
5	40.9%	18.2%	22.7%	18.2%		37.0%	30.4%	19.0%	13.7%

For each class, Table 2 shows the percentage of samples and the corresponding traffic volume for H within specific intervals when submitted to the variance analysis. The analysis of these tables demonstrates that distinct classes can behave very differently. Note that most of the class 3 traffic volume is included in these samples. Class 3 (HTTP traffic) is clearly the one showing the higher degree of burstiness. Most of the samples (76%) exhibit an $H > 0.5$, and 55% of them have an $H > 0.7$. Table 3 extends this analysis to the activity periods defined in section 3.2. Similar analysis was also carried out for the other classes.

Table 3. Class 3: Percentage of samples and traffic volume with an estimate $H > 0.5$.

	Perc. of Samples			Traffic Volume	
Network activity	$0.5 < H < 0.7$	$H \geq 0.7$		$0.5 < H < 0.7$	$H \geq 0.7$
High	14.3%	78.6%		4.0%	95.5%
Medium	33.3%	66.7%		1.6%	98.4%
Low	22.2%	11.1%		25.6%	14.3%

It is notorious that H increases with network activity, which is consistent with [7]. Although the above tables do not differentiate the results by interface, the analysis of classified traffic per interface shows the same tendency. Excluding class 2, the relation between H and the traffic volumes is not clear for the remaining classes which may indicate an application type dependence. In fact, class 1 behaves in opposite way, and class 4 does not show evidence of burstiness for the different activity periods. This can be due to the regular nature of traffic it emcompasses (e.g. routing traffic). As regards the autocorrelation, almost all the samples with autocorrelation functions decaying slowly to zero (which suggests LRD) had an H above 0.5 in the variance time plots, which is consistent.

5 Conclusions

In this study, the Hurst parameter was used to measure LRD in real traffic samples classified according to a proposed criterion. The values for H were determined for the defined traffic classes and for periods of different network activity.

The results show that classes 2 and 3 (bulk transfer and HTTP traffic) play a major role in the total load per interface. In particular, Class 3 is clearly the one showing a higher evidence of burstiness, which increases with traffic load. While Class 2 has similar characteristics, Class 1 behaves in opposite way. Class 4 presents an estimated H below 0.5 independently of the network activity period.

For most of the samples, the tests illustrate similar results either analysing the time series of packets or bytes.

Currently, a larger set of samples are being analysed to consolidate these results. Obtaining complementary statistics is also a matter of concern.

References

1. S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. An Architecture for Differentiated Services. Technical report, IETF RFC 2475, 1998.
2. M.S. Taqqu, W. Willinger, W.E. Leland, and D.V. Wilson. On the Self-Similar Nature of Ethernet Traffic. *SIGCOMM'93*, 1993.
3. Cisco Systems. NetFlow. http://www.iagu.on.net/software/netflow-collector.
4. V. Jacobson, K. Nichols, and K. Poduri. An Expedited Forwarding PHB. Technical report, IETF RFC 2598, 1999.
5. J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski. Assured Forwarding PHB Group. Technical report, IETF RFC 2597, 1999.
6. A. Erramilli and W. Willinger. Experimental Queueing Analysis with Long-Range Dependent Packet Traffic. *IEEE/ACM Trans. on Networking*, 4(2), April 1996.
7. M.S. Taqqu, V. Teverovsky, and W. Willinger. Estimators for Long-Range Dependence: An Empirical Study. *Fractals*, 3:785...788, 1995.
8. Trends in Wide Area IP Traffic Patterns. http://www.caida.org/outreach/papers/.
9. A. Bak, W. Burakowski, F. Ricciato, S. Salsano, and H. Tarasiuk. Traffic Handling in AQUILA QoS IP Networks. *QoFIS2001*, page 243...260, September 2001.
10. A. Croll and E. Packman. *Managing Bandwidth: Deploying QoS in Enterprise Networks.* Prentice Hall, 2000.

Improved Initial Synchronisation in the Presence of Frequency Offset in UMTS FDD Mode

Valentina Lomi[1], Gianfranco L. Pierobon[2], Daniele Tonetto[1], and Lorenzo Vangelista[1]

[1] Telit Mobile Terminals S.p.A.,
via Masini 2, 35129 Padova, Italy
{valentina.lomi, daniele.tonetto, lorenzo.vangelista}@telital.com
[2] Università di Padova, Dipartimento di Elettronica e Informatica,
via Gradenigo 6/A, 35131 Padova, Italy
gianfranco.pierobon@unipd.it

Abstract. The UMTS–FDD system, one of the members of the ITU IMT–2000 third generation standard for terrestrial cellular systems, employs pruned Golay sequences to enable initial synchronisation of the mobile terminals to the network. In this paper a low complexity solution for initial synchronisation is proposed, which is able to counteract the performance degradation introduced by large frequency offsets occurring in the mobile station receiver. Simulation results are provided to validate the proposed solution.

Keywords: cell search, UMTS, FDD, Golay sequences, synchronisation

1 Introduction

The initial synchronisation is the process of a mobile station in a cellular CDMA network getting synchronised in time to the (strongest) base station and acquiring the scrambling code that base station uses for the downlink traffic channels.

To let the mobile get synchronised to the network, the UMTS–FDD system [1] [2] provides two "bursty" pilot channels ("primary" and "secondary" synchronisation channels) and a continuous pilot channel. In this paper we focus on the "primary" channel which is based on the repetition of a non–scrambled Golay sequence common to all cells and which is needed to perform slot synchronisation (the first step of the initial synchronisation procedure, see [3]).

During the standardisation process particular attention has been paid to the necessity of an implementation with requirements of low complexity and of robustness to the frequency offsets, as low cost mobile stations in UMTS systems may have large initial frequency offsets, up to 26 kHz [4]. In [4] [5] and [6], algorithms which are able to counteract the effect of the frequency offset on the initial synchronisation procedure are presented.

In this paper we propose another solution for slot synchronisation in the presence of a frequency offset, which performs better than [4] in most cases. The theoretical basis for our solution is a theorem, proven in this paper, according

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 1165–1171, 2002.

to which the Golay sequences rotated at the receiver by a frequency offset still preserve the “Golay property”.

The paper is organised as follows. Section 2 describes the slot synchronisation procedure in the UMTS–FDD system. Section 3 demonstrates the previously mentioned theorem. Section 4 describes the proposed algorithm and its performance. Conclusions are drawn in Section 5.

2 System Model

For purposes of the slot synchronisation (see [1] for full details), we model the baseband representation of the signal received and sampled at the mobile station at chip rate $1/T_c$, $T_c = 1/3840000$s, as

$$x_r(kT_c) = A_{ch} \cdot e^{\jmath(2\pi \Delta f kT_c + \theta)} \cdot p_{SCH}\left(\left((k-k_0) \bmod Q\right) T_c\right) \ + \tilde{w}(kT_c) \tag{1}$$

where

- $Q = 2560$; QT_c is the time interval called *slot* in the UMTS specifications;
- A_{ch} is a real value modelling a constant channel attenuation;
- Δf is the receiver frequency offset and θ is an unknown phase;
- $k_0 T_c$ is the time offset (unknown to the receiver): its estimation is actually the target of slot synchronisation;
- $\tilde{w}(kT_c)$ is white Gaussian noise with variance σ^2;
- $p_{SCH}\left((k \bmod Q) T_c\right)$ is the *primary synchronisation channel*, common to all UMTS base stations. The sequence

$$p_{SCH}(kT_c) = \begin{cases} \frac{1}{\sqrt{2}} \frac{1}{\sqrt{256}} (1 + \mathrm{j}) \ a(k) & \text{for} \quad 0 \le k < 256 \\ 0 & \text{for} \quad 256 \le k < Q \end{cases} \tag{2}$$

repeats every slot. $a(k)$ is a pruned Golay complementary sequence generated as follows:

$$a_0(k) = \delta(k) \qquad b_0(k) = \delta(k) \tag{3}$$

$$a_n(k) = a_{n-1}(k) \ + \ W_n \cdot b_{n-1}\left(k - D_n\right) \tag{4}$$

$$b_n(k) = \begin{cases} a_{n-1}(k) \ - \ W_n \cdot b_{n-1}\left(k - D_n\right) & n = 1,2,3,5,7,8 \\ a_n(k) & n = 4,6 \end{cases} \tag{5}$$

$$a(k) = a_8(k) \tag{6}$$

with $k = 0, 1, 2, \ldots, 255$, where

$$[D_1, D_2, D_3, D_4, D_5, D_6, D_7, D_8] = [128, 64, 16, 32, 8, 1, 4, 2] \tag{7}$$

$$[W_1, W_2, W_3, W_4, W_5, W_6, W_7, W_8] = [1, -1, 1, 1, 1, 1, 1, 1] \tag{8}$$

Let CNR be the ratio between the power of the primary synchronisation channel and the power of noise, hence $CNR = A_{ch}^2 / \left(256\sigma^2\right)$.

An efficient estimation of k_0 can be obtained with the *Budisin correlator* (see [7]) shown in Fig.1, applied with the following procedure:

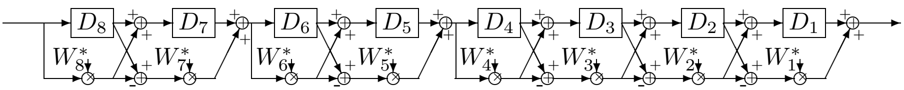


Fig. 1. *Budisin correlator*

1. letting L be the number of slots in a synchronisation time, calculate
$$m(n) = \sum_{\ell=0}^{L-1} \left|c\left(\left(n+\ell Q\right)T_c\right)\right| \qquad n = 0, 1, \ldots, Q-1$$
where $c\left(kT_c\right)$ is the signal at the output of the correlator;
2. find n_0 such that $m(n_0) \geq m(n)$, $n = 0, 1, \ldots, Q-1$;
3. calculate $\hat{k}_0 = n_0 - 255$.

3 Rotated Golay Sequences

Theorem 1. *If the sequence $a(k)$ is a pruned Golay complementary sequence defined by the recursive equations (3), (4), (5) and (6), then the sequence $\hat{a}(k) = a(k)e^{\jmath 2\pi \Delta f k T_c}$ is a pruned Golay complementary sequence too (from now on called rotated Golay sequence), which can be obtained from the recursive equations (3), (4), (5), and (6) with the substitution $W_n \mapsto \hat{W}_n = W_n \cdot e^{\jmath 2\pi \Delta f D_n T_c}$.*

Proof. Equations (3), (4), (5) and (6) can be rewritten in the z–transform domain as

$$A_n(z) = A_{n-1}(z) + W_n\, z^{-D_n}\, B_{n-1}(z) \tag{9}$$

$$B_n(z) = \begin{cases} A_{n-1}(z) - W_n\, z^{-D_n}\, B_{n-1}(z) & \text{for } n = 1, 2, 3, 5, 7, 8 \\ A_n(z) & \text{for } n = 4, 6 \end{cases} \tag{10}$$

$$A(z) = A_8(z) \tag{11}$$

with the initial condition $A_0(z) = B_0(z) = 1$. Defining

$$\hat{a}_n(k) = a_n(k)e^{\jmath 2\pi \Delta f T_c} \tag{12}$$

$$\hat{b}_n(k) = b_n(k)e^{\jmath 2\pi \Delta f T_c} \tag{13}$$

and substituting $z \mapsto z e^{-\jmath 2\pi \Delta f T_c}$ in (9), (10) and (11), we have that the sequence $\hat{a}(k)$ can be represented, in the z–transform domain, by the recursive equations

$$\hat{A}_n(z) = \hat{A}_{n-1}(z) + W_n e^{\jmath 2\pi D_n \Delta f T_c}\, z^{-D_n}\, \hat{B}_{n-1}(z) \tag{14}$$

$$\hat{B}_n(z) = \begin{cases} \hat{A}_{n-1}(z) - W_n e^{\jmath 2\pi D_n \Delta f T_c}\, z^{-D_n}\, \hat{B}_{n-1}(z) & \text{for } n = 1, 2, 3, 5, 7, 8 \\ \hat{A}_n(z) & \text{for } n = 4, 6 \end{cases} \tag{15}$$

$$\hat{A}(z) = \hat{A}_8(z) \tag{16}$$

with the initial conditions $\hat{A}_0(z) = \hat{B}_0(z) = 1$.

4 The Proposed Synchronisation Algorithm and Its Performances

It is known that the usual synchronisation procedure, described in Section 2, is very sensitive to frequency offsets. It can be shown (see [4]) that the *signal-to-noise* ratio degradation in the output of the correlator matched to the sequence $a(k)$ is proportional to

$$\frac{\sin^2 N\pi\Delta f T_c}{N\sin^2 \pi\Delta f T_c} \tag{17}$$

with $N = 256$. Hence the correlation peak vanishes at all when $\Delta f = \Delta f_\pm = \pm 1/NT_c = \pm 15$ kHz.

This consideration together with Theorem 1 lead us to propose the new solution, depicted in Fig. 2, which uses three *Budisin correlators*, one matched to the sequence $a(k)$, one matched to $a(k) \cdot e^{\jmath 2\pi\Delta f_+ kT_c}$ and one matched to $a(k) \cdot e^{\jmath 2\pi\Delta f_- kT_c}$. [1] The *MAX* block takes the sequences applied at its input, determines the maximum in the set of all the values assumed by the sequences and produces as an output the input sequence to which the maximum belongs.

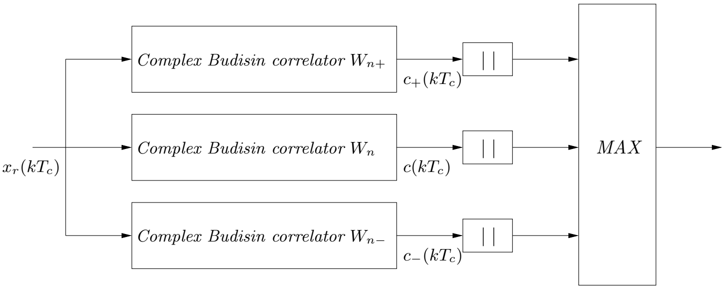

$x_r(kT_c)$
Complex Budisin correlator W_{n+}
Complex Budisin correlator W_n
Complex Budisin correlator W_{n-}
$c_+(kT_c)$
$c(kT_c)$
$c_-(kT_c)$
| |
| |
| |
MAX

Fig. 2. The proposed algorithm

Unfortunately the proposed synchronisation scheme has a higher implementation complexity than usual schemes because the coefficients $W_{n\pm}$ are $e^{\jmath 2\pi\Delta f_\pm D_n T_c}$ instead of simply ±1 and three correlators, instead of only one, are used.

In order to reduce the implementation complexity we make the following approximations for the coefficients of the upper and lower correlators:

$$W_{m+} \approx W_m \tag{18}$$

$$W_{m-} \approx W_m \qquad \text{for } m = 3, 4, 5, 6, 7, 8 \tag{19}$$

[1] Note that at the critical frequency offsets 0, Δf_+ and Δf_- one of the outputs of the three correlators raises its maximum, while the others have there their minima.

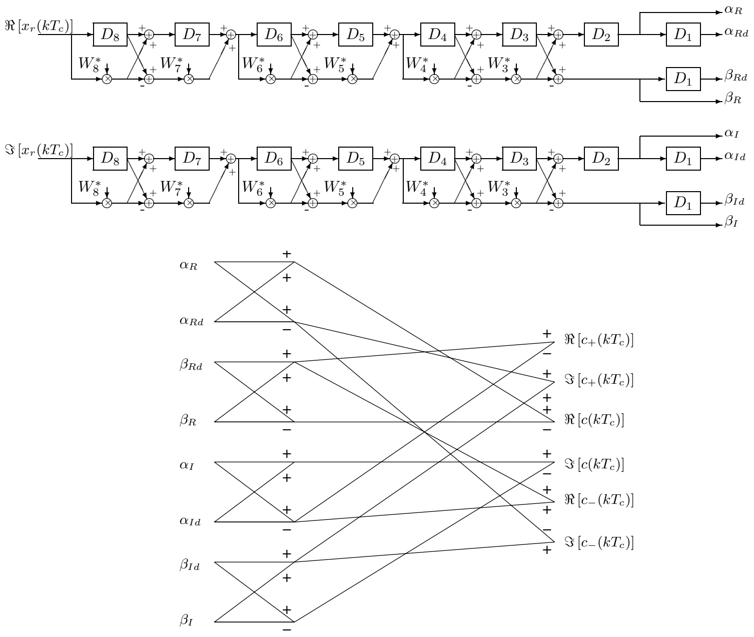


Fig. 3. Reduced complexity scheme

It can be demonstrated that these approximations have the correlators matched to sequences rotated by a staircase phase instead of a linear increasing phase. Taking into account the above assumptions, computations depicted in Fig. 2 can be re–organised as shown in Fig. 3 in a reduced complexity scheme. Note that, while the single Budisin correlator (considering both in–phase and quadrature components) needs 26 sums per output, the computations required by the proposed algorithm in this simplified implementation are 34 sums per output.

The performance of the proposed algorithm in a flat fading channel with 9.26 Hz Doppler is shown in Fig. 4 with no frequency offset and with a 20 kHz frequency offset. In both cases it is compared with the performances obtained with a single Budisin correlator and with the algorithm described in [4] (indicated with the label "Wang-Ottosson"). The algorithm proposed in this paper shows a very good performance both at 0 kHz and 20 kHz. The synchronisation error is also plotted versus the frequency offset in Fig. 5 for both the algorithm in [4] (indicated with the label "W-O") and the proposed method. Examined CNR values are -21 dB and -13 dB. The proposed algorithm shows the best behaviour for all the examined frequency range, except for the values around $\Delta f_+/2$.

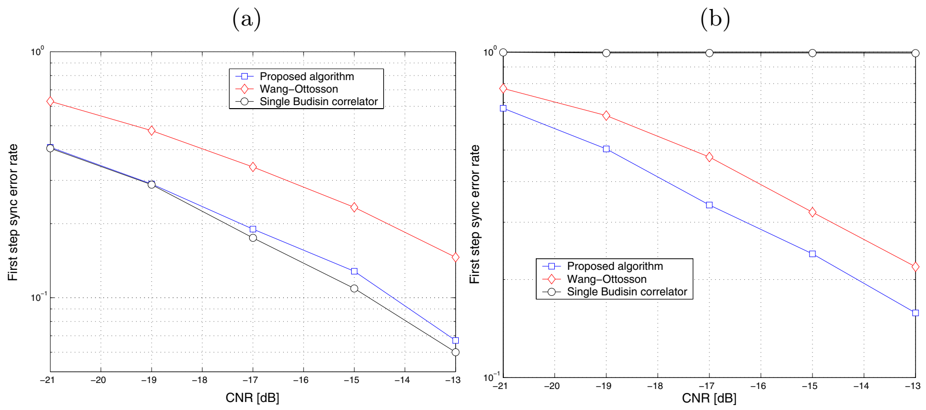


Fig. 4. First step performance with a 0 (a) and 20 kHz (b) frequency offset ($L = 15$)

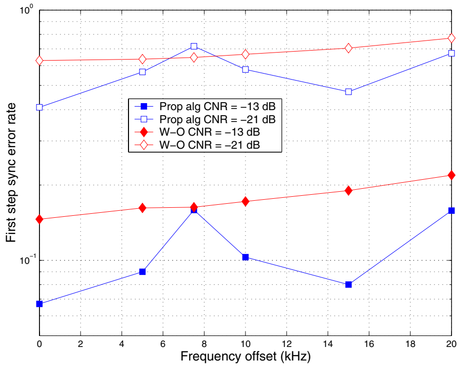


Fig. 5. First step performance in two different CNR conditions ($L = 15$)

5 Conclusions

We have presented an innovative method for the first step of the UMTS–FDD initial synchronisation procedure which is able to counteract the degrading effect of the frequency offset. The algorithm has a low complexity implementation. Simulations indicate that it can offer a good performance for all the CNR and frequency offset values of interest.

References

1. 3GPP *3G TS 25.211 "Physical channels and mapping of physical channels onto transport channels (FDD)"*, version 4.1.0 June 2001.
2. 3GPP *3G TS 25.213 "Spreading and modulation (FDD)"*, version 4.1.0 June 2001.
3. 3GPP *3G TS 25.214 "Physical layer procedures (FDD)"*, version 4.1.0 June 2001.

4. Y.-P. E. Wang, T. Ottosson, "Cell search in W–CDMA", *IEEE Journal on Selected Areas in Communications*, Vol. 18, No. 8, Aug. 2000.
5. K.-M. Lee, J.-Y. Chun, "An initial cell search scheme robut to frequency error in W-CDMA system", *PIMRC 2000*, Vol. 2, 2000.
6. S.-Y. Hwang, B.-J. Kang, J.-S. Kim, "Performance analysis of initial cell search using time tracker for W-CDMA", *GLOBECOM 2001*, Vol. 5, 2001.
7. S. Z. Budisin, "Efficient pulse compressor for Golay complementary sequences", *Electronics Letters*, Vol. 27, No. 3, Jan. 1991.

Scalable Adaptive Hierarchical Clustering*

Laurent Mathy[1], Roberto Canonico[2], Steven Simpson[1], and David Hutchison[1]

[1] Lancaster University, UK
{laurent, ss, dh}@comp.lancs.ac.uk
[2] University Federico II, Napoli, Italy
roberto.canonico@unina.it

Abstract. We propose a new application-level clustering algorithm capable of building an overlay spanning tree among participants of large multicast sessions, without any specific help from the network routers. This algorithm is based on a unique definition of zones around nodes and an innovative adaptive cluster size distribution. The proposed method finds application in many context where large-scale overlay trees can be usefull: application-level multicasting, peer-to-peer networks and content distribution networks (among other things).

1 Introduction

More than a decade of research in multicast technologies demonstrates the need for large-scale (application-level) overlay structures.

Tree-based ACKs (TRACKs) have been identified as the most appropriate approach to providing real-time and scalable delivery guarantees to groups of receivers [8]. In this scenario, the overlay provides a control structure. This approach is further re-inforced with the recent emergence of the Source-Specific IP multicast model [4][3] which is an asymmetrical service where only a designated source can send in multicast to the group.

More recently, reasons for the lack of widespread deployment of IP multicast have been identified [2]. These indicate that ubiquitous rollout of IP multicast services may, even if at all possible, take a very long time. In such circumstances, overlays represent an attractive alternative to IP multicast for data dissemination among members of multicast groups. This is the case in Content Delivery Networks (CDN) where application-level multicast overlays are often used, for example, for the distribution of multimedia data from primary to secondary servers.

Peer-to-peer (p2p) applications also rely heavily on overlays. Here, the overlays are used to propagate search strings among the nodes of the p2p network, in order to discover the location of the desired content. Very often, users of p2p networks statically configure a few nodes to peer with, which can result in a non-efficient, almost "chaotic" overlay.

To date, all the above scenarii lack, but would greatly benefit from, effective algorithms to build large-scale, efficient overlays. In this paper, we propose a

* This work was supported by the BT Alpine Project.

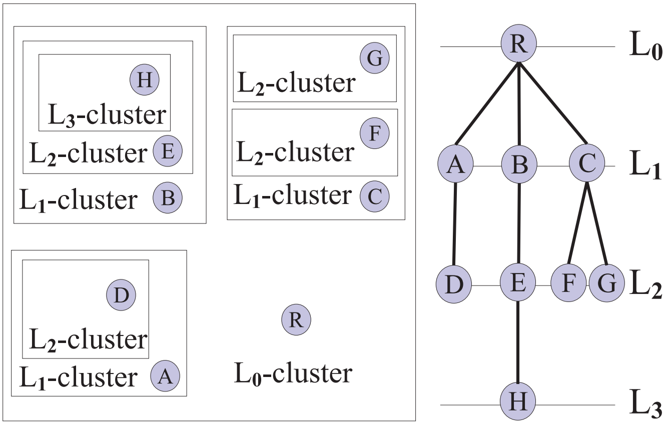
E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 1172–1177, 2002.

new method designed to build such large-scale overlays, without requiring any special support from the network routers. This method is based on the concept of clustering.

2 Adaptive Hierachical Clustering Algorithm

2.1 General Strategy and Goal

The algorithm described in this section is designed to build, recursively, a hierarchy of *clusters*. A cluster is represented by a *cluster head* and is composed of the cluster head and other nodes "close" to the cluster head. The algorithm is "recursive" in the sense that each cluster is divided into sub-clusters, whose (sub-)cluster heads are constituent nodes of the original cluster. The hierarchy of clusters is organised into *layers*, where layer L_i is composed of the cluster heads of (sub-)clusters that divide L_{i-1}-clusters (i.e. clusters whose head is in layer L_{i-1}). This is illustrated in figure 1.(a). For instance, in this figure, the L_1-cluster headed by C is composed of C, F and G. This cluster contains two L_2-clusters, respectively headed by F and G.



1.(a): Clusters and layers 1.(b): Tree

Fig. 1. Cluster hierarchy.

The principle of the algorithm is that, starting at layer L_0 with a top-level cluster containing all the nodes in the hierarchy and whose cluster head is a well known node called the *root* of the hierarchy, clusters are recursively divided into sub-clusters, until all clusters obtained are “singleton-clusters” containing only their cluster head[1].

The cluster hierarchy thus built forms a logical tree spanning all the cluster heads (e.g. all the nodes) in the hierarchy (see figure 1.(b)). Consequently, the state to build and maintain this hierarchy can be distributed among all the nodes in the hierarchy such that each node in layer L_i only needs to record its parent cluster head (i.e. the L_{i-1}-cluster head whose cluster it belongs to) and the L_{i+1}-cluster heads that are members of its own cluster. For instance, in figure 1.(a), B records R as its parent cluster and E as its child.

2.2 Workings of the Algorithm

The algorithm is distributed and based solely on unicast communications. In other words, it does not rely on any special network support.

One of the central ideas in the algorithm is that any node (i.e. any cluster head) sees the rest of the world as a set of concentric rings (which we call *zones*), centered on the node itself. Each zone starts where the previous one finishes and the zones are numbered in increasing order, starting at 0 for the smallest ring (see figure 2). The actual size of each ring, as well as the distance measurement used to define it (e.g. delays, throughput, etc.), is unimportant for the general workings of the algorithm. With each zone, a distance called a *radius* is also defined. Again, the size of the radius is unimportant for the workings of the algorithm (but its distance measurement is the same as the measurements used for defining the zones).

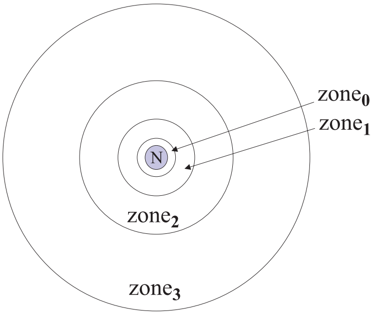


Fig. 2. Zones associated with a node.

The scalable hierachical clustering algorithm works as follows. The cluster hierarchy is rooted on a well known entity called the *root*. A node desiring to join the cluster hierarchy first measures its distance to the root, and then sends

[1] Each node in the hierarchy is therefore the cluster head of a (sub-)cluster.

to the root a JOIN message containing this distance. Based on this distance, the root determines the zone of the joining node. Here, two cases are possible:

1. The joining node is the first node joining in the corresponding zone.
2. Other nodes from the same zone have already joined.

In the former case, the root records the presence of the joining node in the corresponding zone and sends the node a NEW_CLUSTER_ACK message, indicating that the joining node has found its place in the hierarchy (this finishes the algorithm for the joining node). The joining node is now the cluster head of one of the sub-clusters dividing the cluster headed by the root (albeit a "singleton-cluster" for the time being).

In the latter case, the root sends to the joining node, in a TRY message, the list of the cluster heads in the same zone as the joining node, along with the radius associated with this zone. The joining node then measures its distance to each of the nodes in the list. Again, we consider two cases:

1. The distance of the joining node to at least one of the cluster heads in the list is smaller than the given radius. The clusters headed by these cluster heads are called *attracting clusters*.
2. The distance of the joining node to all the cluster heads in the list is greater than the given radius.

In the former case, the joining node chooses the closest attracting cluster and joins it: that is, the algorithm starts again with the corresponding cluster head acting as the root. In this case, the joining node is said to "go down one layer" (as the cluster it is heading will potentially be part of the partition of the attracting cluster) and it is important to note that the root does not record the presence of the joining node. In essence, from the root's point of view, the members of the attracting cluster are "collapsed" into the attracting cluster head, as this cluster head is the only node in the attracting cluster remembered by the root.

In the latter case, the joining node creates a new sub-cluster by sending a NEW_CLUSTER message to the root (including its distance to the root). The root then keeps a record of the new cluster head (i.e. the joining node) and of its zone and replies with a NEW_CLUSTER_ACK message which finishes the algorithm for this joining node.

3 Scalability Considerations

From the previous section, it should be clear that the state overhead imposed on each node in the hierarchy is proportional to the number of zones needed for that node to "span" its cluster, times the number of clusters per zone. This number of clusters per zone also influences the scalability of the join procedure, as any joining node must measure its distance to all the cluster heads at the same zone, for all traversed layers. Also, the further away from the central node a zone is, the more nodes – and thus the more clusters – such a zone potentially contains. These observations favour the use of large clusters, within few zones.

On the other hand, large clusters tend to create many layers (as they can contain large sub-clusters which, in turn, will have to be divided), which has a negative impact on the latency of the join procedure.

In order to accommodate these conflicting requirements, we propose to define zones based on RoundTrip Times (RTTs) measurements, and whose sizes follow an "exponential distribution" (see figure 2):

$$\text{zone}_0 : 0 < \text{dist} \leq 1 \tag{1}$$

$$\text{zone}_i : (1 + \mathit{\Delta})^{i-1} < \text{dist} \leq (1 + \mathit{\Delta})^i, \text{with } \mathit{\Delta} > 0 \tag{2}$$

This, in turn, allows us to define the size (i.e. radius) of the clusters at zone$_i$ as:

$$r_i = \frac{(1 + \mathit{\Delta})^i - (1 + \mathit{\Delta})^{i-1}}{2} \tag{3}$$

The parameter $\mathit{\Delta}$ in the formulae could be either fixed or varied according to which layer the cluster, headed by the corresponding node, belongs to. Other size "distributions" for both zones and radii are of course possible, but the ones we propose prevent an explosion of the number of clusters in far zones while keeping the number of zones down and retaining the desirable property that "detail" (i.e. "fine grain positioning") matters only for nodes close to a cluster head.

4 Discussion and Conclusions

In this paper, we have proposed a method to build a hierarchy of nodes, based on the notion of proximity, in a distributed and scalable way. The hierachy is built through a series of "local" decisions involving only a small subset of the hierachy's population for each decision. This, coupled with an innovative adaptive cluster size distribution approach, yields a simple, yet powerful, approach to building overlay, application-level structures without relying on any special support from network routers.

The hierarchy thus built is loopless and spans all the nodes in it. Our scalable adaptive hierarchical clustering algorithm can therefore be seen as a new member in the category of application-level multicast tree building methods (e.g. [1][5][7][6]). The overlay application-level multicasting trees built with our scalable adaptive hierarchical clustering are unconstrained, meaning that nodes in the tree cannot explicitly control their number of children. This may not be a problem for overlay trees built for control purposes [8] but could yield a significant penalty for trees built for data distribution. However, the method presented in this paper can still be very useful in the context of application-level multicast data distribution.

Indeed, a constrained application-level multicast overlay tree can be built by having each cluster head and its sub-clusters (i.e. its members populating the next layer in the hierarchy) run any algorithm that builds a constrained overlay tree [1][7][6]. With this approach, each node in the cluster hierachy would be a member of the overlay tree rooted at its parent cluster, as well as the root of the overlay tree spanning its own cluster. This would allow the building of very large constrained overlay trees.

Another application of the scalable adaptive hierarchical clustering presented in this paper is resource discovery. Indeed, a permanent hierarchy of resources could be built, rooted on a well known node, and "searched" by clients with a modified join procedure which does not declare the creation of a new (sub-) cluster when it finishes (see section 2.2). This could even substitute *expanding ring searches* in asymmetric network multicast circumstances or when network multicast is unavailable.

In future work, we will investigate the performance of the proposed algorithm under dynamic conditions (e.g. dynamic group membership, failures, etc.)

References

1. Y-H. Chu, S. Rao, and H. Zhang. A Case for End System Multicast. In *ACM SIGMETRICS*, pages 1–12, Santa Clare, CA, USA, June 2000.
2. C. Diot, B. Levine, B. Lyles, H. Kassem, and D. Balensiefen. Deployment Issues for the IP Multicast Service and Architecture. *IEEE Network*, 14(1):78–88, Jan/Feb 2000.
3. B. Fenner, M. Handley, H. Holbrook, and I. Kouvelas. Protocol Iindependent Multicast - Sparse Mode: Protocol Specification (revised). Internet Draft draft-ietf-pim-sm-v2-new-02, IETF, Mar 2001. Work in Progress.
4. H. Holbrook and D. Cheriton. IP Multicast Channels: EXPRESS Support for Large-scale Single-source Applications. *ACM Comp. Comm. Reviews*, 29(4):65–78, Oct 1999.
5. J. Jannotti, D. Gifford, K. Johnson, F. Kaashoek, and J. O'Toole. Overcast: Reliable Multicasting with an Overlay Network. In *USENIX OSDI*, San Diego, CA, USA, Oct 2000.
6. L. Mathy, R. Canonico, and D. Hutchison. An Overlay Tree Building Control Protocol. In *Proc. of Intl. workshop on Networked Group Communication (NGC)*, pages 76–87, Nov 2001.
7. D. Pendarakis, S. Shi, D. Verma, and M. Waldvogel. ALMI: an Application Level Multicast Infrastructure. In *3rd USENIX Symposium on Internet Technologies*, San Francisco, CA, USA, Mar 2001.
8. B. Whetten and G. Taskale. An Overview of Reliable Multicast Transport Protocol II. *IEEE Network*, 14(1):37–47, Jan 2000.

How to Achieve Fair Differentiation

Eeva Nyberg and Samuli Aalto

Networking Laboratory
Helsinki University of Technology
P.O.Box 3000, FIN-02015 HUT, Finland
{eeva.nyberg,samuli.aalto}@hut.fi

Abstract. We present a simple packet level model to show how marking at the DiffServ boundary node and scheduling and discarding inside a DiffServ node affect the division of bandwidth between two delay classes: elastic TCP flows and streaming non-TCP flows. We conclude that only per flow marking together with dependent discarding thresholds across both delay classes is able to divide bandwidth fairly, according to the load of the network, and in a TCP friendly way.

Keywords: DiffServ, TCP, fairness, TCP friendliness

1 Introduction

The main arguments against differentiation are the waste of network resources and the difficulty to guarantee fair bandwidth allocation between priority classes. More research in this field has to be done, to be able to settle the dispute. The Internet research also lacks efforts in coupling the packet level QoS mechanisms of DiffServ [1], e.g. Assured Forwarding (AF) [2], to flow level analysis. On the other hand, flow level bandwidth allocation and fairness research, e.g. [3], [4], continue to assume that weighted fair bandwidth allocations between flows in different service classes are somehow achieved and evade the question of how to do so without flow control or per flow scheduling.

In [5] we introduced both packet and flow level models to study how bandwidth is divided among flows using packet level differentiation mechanisms of the Simple Integrated Media Access (SIMA) proposal [6]. In the present paper we continue the packet level modelling approach to investigate the key factors of two DiffServ schemes, AF and SIMA. Following Roberts [7], we assume two forwarding classes based on delay requirements: elastic TCP traffic and streaming non-TCP traffic. As a result, we present the role of the conditioning and forwarding mechanisms in dividing bandwidth consistently across delay classes.

2 DiffServ Network Model and Its Analysis

The main elements of DiffServ are traffic classification and conditioning at the boundary nodes and traffic forwarding through scheduling and discarding at the DiffServ interior nodes. In addition, congestion control mechanisms designed for

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 1178–1183, 2002.

the Internet, such as TCP, and active queue management algorithms, such as RED, may be used for QoS in the Internet. Figure 1 summarizes the components.

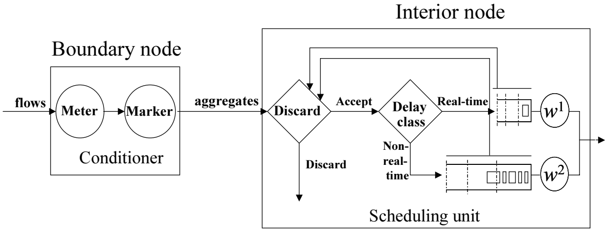


Fig. 1. Components of a DiffServ network

Network model. Consider a DiffServ network with a single bottleneck link, which is loaded by a fixed number of flows. Assume two delays classes, $d = 1, 2$, and I precedence levels, $i = 1, \ldots, I$. Delay class 1 refers to non-TCP flows, and delay class 2 to TCP-flows. Precedence level I refers to the highest priority, i.e. flows at that level encounter the smallest packet loss probability, and level 1 to the lowest priority. Note that this is just opposite to, e.g., the definition given in [2]. Therefore, we rather use the term priority level here.

Each flow is given a weight ϕ that reflects the value of the flow. A natural objective of any traffic control algorithm is to allocate bandwidth as fairly as possible. Here fairness refers to weighted fairness in a single link, i.e. the throughput θ of any flow should be proportional to its weight ϕ. For networks with DiffServ architecture it is not clear how to achieve this objective, since there are no per flow mechanisms available in the core network.

At the conditioner, the packets of a flow are marked to priority levels according to the measured traffic rate compared to the weight of the flow. More specifically, let ν denote the measured packet arrival rate of a flow. As in [6], we assume that the priority level pr of the flow depends on ν and ϕ as follows:

$$pr = \max\left[\min\left[\left\lfloor I/2 + 0.5 - \frac{\ln \frac{\nu}{\phi}}{\ln 2}\right\rfloor, I\right], 1\right]. \tag{1}$$

Thus, the priority level is decreased by one as soon as the traffic rate doubles.

For non-TCP flows we assume a fixed packet arrival rate ν, whereas for TCP flows it depends on the congestion level of the network. Let RTT denote the round trip time of a TCP flow and q the packet loss probability it encounters in the buffer of the bottleneck link. Following [8], we assume that

$$\nu = \frac{1}{RTT}\sqrt{2\frac{1-q}{q}}. \tag{2}$$

Assume that there are L^1 different groups of non-TCP flows, each group l with a characteristic packet arrival rate $\nu(l)$, and let $\mathcal{L}^1$ denote the set of such flow groups. Furthermore, assume that there are L^2 different groups of TCP flows, each group l with a characteristic round trip time $RTT(l)$, and let $\mathcal{L}^2$

denote the set of such flow groups. Finally, let $n(l)$ denote the number of flows in any group l.

At the boundary node all the traffic belonging to the same delay class and precedence level are aggregated. Let $\lambda^d(i)$ denote the aggregate packet arrival rate of delay class d and priority level i. Packets of the flow aggregates are then forwarded or discarded by a scheduling unit that includes two buffers, one for each delay class. Denote by K^1 and K^2 the sizes of the two buffers in number of packets.

DiffServ mechanisms. Traffic is conditioned at the boundary node by measuring the incoming traffic and, based on the metering result, by marking the packets of the flow. We consider two different **marking principles**:

– *Per flow marking:* Once the measured traffic rate of a flow exceeds a marking threshold, all packets of the flow are marked to the same precedence level.
– *Per packet marking:* Only those packets of a flow that exceed the marking threshold are marked to the lower precedence level.

The marking thresholds for flow group l, determined from (1), are $t(l,0) = \infty$, $t(l,I) = 0$, and

$$t(l,i) = \phi(l) \cdot 2^{I/2-i-0.5}, \ i = 1,...,I-1. \tag{3}$$

Per flow marking gives the aggregate arrival intensity $\lambda^d(i)$ as

$$\lambda^d(i) = \sum_{l \in \mathcal{L}^d : pr(l)=i} n(l)\nu(l). \tag{4}$$

On the other hand, if *per packet* marking is applied, then

$$\lambda^d(i) = \sum_{l \in \mathcal{L}^d : pr(l) \le i} n(l)(\min[\nu(l), t(l,i-1)] - \min[\nu(l), t(l,i)]). \tag{5}$$

But what are such **metering and marking mechanisms** that follow these principles? In [9] we demonstrated by simulation experiments that the *token bucket* scheme marks packets to precedence levels *per packet*, while the use of *exponentially weighted moving average* (EWMA) marks packets *per flow*. The token bucket scheme is referred to, e.g., in the AF specification. Packets are marked to I precedence levels by $I-1$ cascaded token buckets. The EWMA scheme was proposed, e.g., in the SIMA proposal.

Forwarding at the interior node is done to aggregates divided, in our case, into two delay classes. Before forwarding, traffic can be limited by discarding packets based on precedence levels. We consider two different **discarding mechanisms**:

– *Independent discarding:* Each buffer acts locally as a separate buffer, discarding appropriate precedence levels according to its buffer content.
– *Dependent discarding:* The content of both buffers determines which precedence level is discarded, in both buffers.

Let m^d denote the number packets in the buffer of delay class d. The independent discarding is implemented by giving, separately for each delay class d, thresholds $K^d(i)$ that determine the minimum priority level accepted, PL_a, when compared to m^d. The dependent discarding, proposed in [6], is implemented by giving a two-dimensional monotonic function

$$PL_a = f(\frac{m^1}{K^1}, \frac{m^2}{K^2}) \tag{6}$$

that determines the minimum priority level accepted when in state (m^1, m^2). We apply the function introduced in [10].

The traffic not discarded is placed in the two buffers. Following the Weighted Fair Queuing (WFQ) principle, whenever one of the buffers is empty, the other buffer has use of total link capacity. Otherwise the capacity of the link is divided according to predetermined weights w^1 and w^2, with $w^1 + w^2 = 1$. We consider three different **scheduling scenarios**:

- *Priority queuing:* WFQ with weights $(w^1 = 1, w^2 = 0)$.
- *Unequal sharing:* WFQ with weights $(w^1 = 0.75, w^2 = 0.25)$.
- *Equal sharing:* WFQ, with weights $(w^1 = w^2 = 0.5)$.

Analysis. The scheduling unit with two buffers is modelled as two dependent $M/M/1/K$ queues with state dependent arrival intensities. When in state (m^1, m^2), the arrival intensity depends on the applied discarding function as follows: if PL_a is i, then the arrival rate for buffer d is $\lambda^d(i)+\ldots+\lambda^d(I)$. The packet transmission times are assumed to be exponentially distributed with mean $1/\mu$. Thus, if both buffers are non-empty, packet service rates are $w^1\mu$ and $w^2\mu$ for the two delay classes. This results in a two-dimensional Markov jump process, the stationary distribution of which can be solved numerically.

From the stationary distribution we can calculate the packet loss probabilities $p^d(i)$ for each traffic aggregate, i.e., for each combination of delay class d and priority level i. Thus, if per flow marking is applied, the packet loss probability, $q(l)$, for a flow in group $l \in \mathcal{L}^d$ becomes

$$q(l) = p^d(pr(l)). \tag{7}$$

On the other hand, if per packet marking is applied, then

$$q(l) = \sum_{j=1}^{I} p^d(j) \frac{\min\left[\nu(l), t(l, j-1)\right] - \min\left[\nu(l), t(l, j)\right]}{\nu(l)}. \tag{8}$$

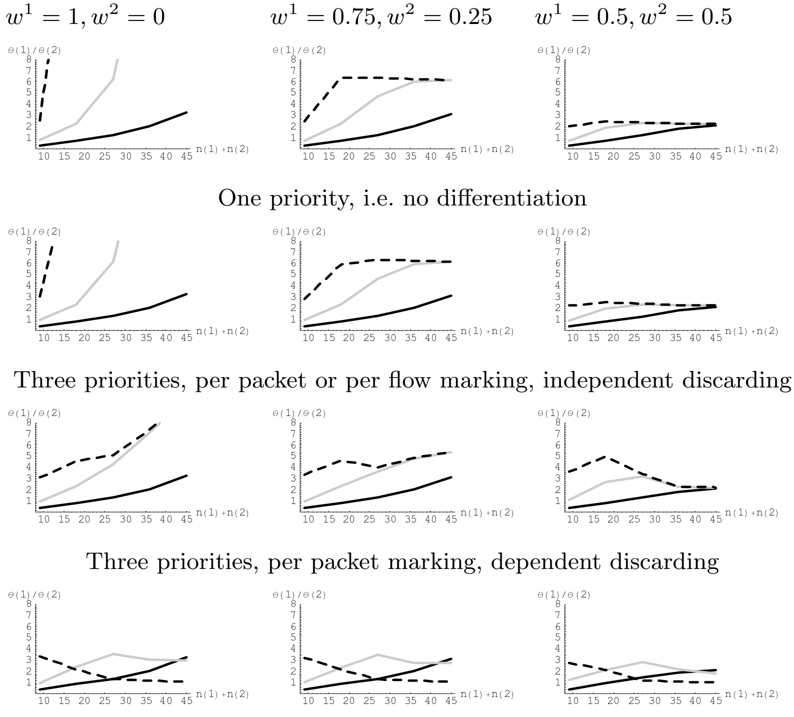
For each TCP flow these packet loss probabilities can be used to determine iteratively the packet arrival rate ν from equation (2). Then these rates are again aggregated as in (4) and (5), and the aggregate rates are used to solve the stationary distribution of the resulting two-dimensional Markov process. By continuing this iteration, the traffic rates of TCP flows converge to some equilibrium values, which reflect the network state, i.e. the number of flows $n(l)$ in different classes l.

3 Numerical Results and Conclusions

We study the combined effect of the three degrees of freedom introduced in the text: marking, discarding thresholds and weighted capacity.

We have the following scenario in terms of the free parameters: $\mu = 1$, $K^1 = 13$, $K^2 = 39$ and $I = 3$. In addition we consider two flow groups, non-TCP flows in group 1 with $\phi(1) = 0.08$ and $\nu(1) \in \{0.039, 0.079, 0.16\}$, and TCP flows in group 2 with $\phi(2) = 0.04$ and $RTT(2) = 1000/\mu$. The three values of $\nu(1)$ are chosen so that, under the per flow marking scheme, the non-TCP flows have priorities $pr(1) = 3$, $pr(1) = 2$, and $pr(1) = 1$, respectively.

Each set of pictures depicted in figure 2 show the ratio $\frac{\theta(1)}{\theta(2)} = \frac{\nu(1)(1-q(1))}{\nu(2)(1-q(2))}$ between throughputs of flows as function of total number of flows, under the condition $n(1)/n(2) = 1/2$. The trajectories are solid, gray, and dashed for $\nu(1) = 0.039$, $\nu(1) = 0.079$, and $\nu(1) = 0.16$, respectively.



$w^1 = 1, w^2 = 0$ $w^1 = 0.75, w^2 = 0.25$ $w^1 = 0.5, w^2 = 0.5$

One priority, i.e. no differentiation

Three priorities, per packet or per flow marking, independent discarding

Three priorities, per packet marking, dependent discarding

Three priorities, per flow marking, dependent discarding

Fig. 2. Effect of marking and discarding when the minimum weights of the rt buffer and nrt buffer change. 66% are TCP flows and 33% non-TCP flows.

The lowest pair in figure 2 shows the effect of per flow marking and dependent discarding. Marking all packets of the flow to the same priority level encourages the TCP mechanism to optimize the sending rate according to the network state. Under congestion, the TCP flows attain a higher priority level by dropping their sending rate. This also encourages the non-TCP traffic to adjust the sending rate accordingly. In all other cases, it is always optimal for the non-TCP flows to send as much as possible, even if packets are then marked to the lowest priority level. The use of per flow marking and dependent thresholds thus gives a powerful incentive for flows to be TCP friendly [11].

The use of dependent discarding controls the throughput of non-responsive flows better than independent discarding. With dependent thresholds, when the nrt buffer is congested packets in the rt buffer are also discarded to alleviate the congestion.

The effect of giving some minimum weight to the nrt buffer protects the TCP traffic from bandwidth exhaustion by the non-TCP flows. However, there is not a clear one to one relationship between the ratio w^1/w^2 of scheduler weights and ratio $\phi(1)/\phi(2)$ of flow group weights.

Further research has to be done in elaborating the TCP congestion control model to include slow start. Furthermore, to properly assess the mechanisms we need to extend the model to networks with more than one bottleneck link.

Acknowledgments. Eeva Nyberg's research is supported by the Academy of Finland and in part by a grant from the Nokia Foundation. The authors would like to thank Jorma Virtamo and Eemeli Kuumola for their cooperation.

References

1. Blake S., Black D., Carlson M., Davies E., Wang Z., and Weiss W., *An Architecture for Differentiated Service*, Dec. 1998, RFC 2475.
2. Heinanen J., Baker F., Weiss W., and Wroclawski J., *Assured Forwarding PHB Group*, June 1999, RFC 2597.
3. Kelly F., "Charging and rate control for elastic traffic," *Eur. Trans. Telecommun*, vol. 8, pp. 33–37, 1997.
4. Massoulié L. and Roberts J., "Bandwidth sharing: Objectives and algorithms," in *Proceedings of IEEE INFOCOM*, 1999, pp. 1395–1403.
5. Nyberg E., Aalto S., and Virtamo J., "Relating flow level requirements to DiffServ packet level mechanisms," Tech. Rep. TD(01)04, COST279, Oct. 2001.
6. Kilkki K., "Simple Integrated Media Access," available at http://www-nrc.nokia.com/sima, 1997.
7. Roberts J., "Traffic theory and the Internet," *IEEE Communications Magazine*, vol. 39, no. 1, pp. 94–99, Jan. 2001.
8. Kelly F., "Mathematical modelling of the Internet," in *Proc. of Fourth International Congress on Industrial and Applied Mathematics*, 1999, pp. 105–116.
9. Nyberg E., Aalto S., and Susitaival R., "A simulation study on the relation of DiffServ packet level mechanisms and flow level QoS requirements," in *Intl. Seminar, Telecommunication Neworks and Teletraffic Theory*, St. Petersburg, Russia, 2002.
10. Laine J., Saaristo S., Lemponen J., and Harju J., "Implementation and measurements of simple integrated media access (SIMA) network nodes," in *Proceedings for IEEE ICC 2000*, June 2000, pp. 796–800.
11. Floyd S. and Fall K., "Promoting the use of end-to-end congestion control in the Internet," *IEEE/ACM Transactions on Networking*, vol. 7, no. 4, pp. 458–472, Aug. 1999.

Measurement-Based Admission Control for Dynamic Multicast Groups in Diff-Serv Networks*

Elena Pagani and Gian Paolo Rossi

Dip. di Scienze dell'Informazione, Università degli Studi di Milano
via Comelico 39, I-20135 Milano, Italy
{pagani,rossi}@dsi.unimi.it

Abstract. An appealing approach to the admission control problem for traffic with QoS requirements consists in evaluating the resource availability by means of *measurement-based* techniques. Those techniques allow to provide QoS with minimal changes to the current network devices. In this work, we propose a mechanism to perform active measurement-based admission control for *multicast groups with dynamically joining receivers*. The proposed mechanism has been implemented in the NS-2 simulation framework, to evaluate its performance.

1 Introduction

The *Differentiated Service* model [1] has been proposed in the literature to provide QoS in a scalable manner. According to that model, *bandwidth broker* agents [4] exist that take in charge the traffic admission control functionalities. Yet, only a few practical implementations of the diff-serv model have been realized. Moreover, in the diff-serv model it is difficult to support multicast [1].

In this paper we describe the end-to-end *Call Admission Multicast Protocol* (CAMP) [5], that can be used to ensure bandwidth guarantees to multicast sessions in IP networks, thus providing them with the Premium Service [4]. CAMP is scalable, operates on a per-call basis and supports the group membership dynamics. It performs the functionalities of a *distributed bandwidth broker* (BB). To perform the admission control, CAMP adopts an active-measurement approach. We have implemented CAMP in the frame of the NS-2 simulation package to verify its effectiveness under different system conditions.

In the system model we consider, a QoS-sensitive application specifies to the underlying service provider, the QoS communication requirements and the behaviors of the data flow it is going to generate (*traffic profile*). We assume that a session announcement protocol (e.g., `sdr`) is available to announce the needed session information. We consider sources generating CBR traffic. All the recipients receive the same set of microflows; they have the same QoS requirements.

* This work was supported by the MURST under Contract no.MM09265173 "Techniques for end-to-end Quality-of-Service control in multi-domain IP networks".

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 1184–1189, 2002.

We adopt the notation proposed in [1] for the *differentiated services* (diff-serv, DS) model. We consider the functional architecture of the BB in accordance with the proposal presented in [7]. The BB provides the applications with the interface to access the QoS services. When the QoS aggregate spans multiple domains, an inter-domain protocol is executed amongst peer BBs, to guarantee the proper configuration of the transit and destination domains.

2 Distributed Bandwidth Broker

In this section we outline the end-to-end *Call Admission Multicast Protocol* (Camp); greater details can be found in [5]. In Figure 1, we show the system architecture in which Camp works. Camp operates within the RTP/RTCP [6] protocol suite and performs the set-up of a RTP session. It receives from the application, via RTP, the profile of the data traffic that will be generated. Camp uses RTCP to monitor the QoS supplied to the recipients. Camp performs the admission control using an *active measurement* approach [2]. It generates *probing* traffic with the same profile as the data traffic generated by the application. Both the data and the `probe` packets are multicast. To this aim, we assume that both a membership protocol and a multicast routing protocol are available. The latter maintains a tree-based routing infrastructure connecting the multicast recipients. Camp is independent of both those protocols. The *probing* phase has

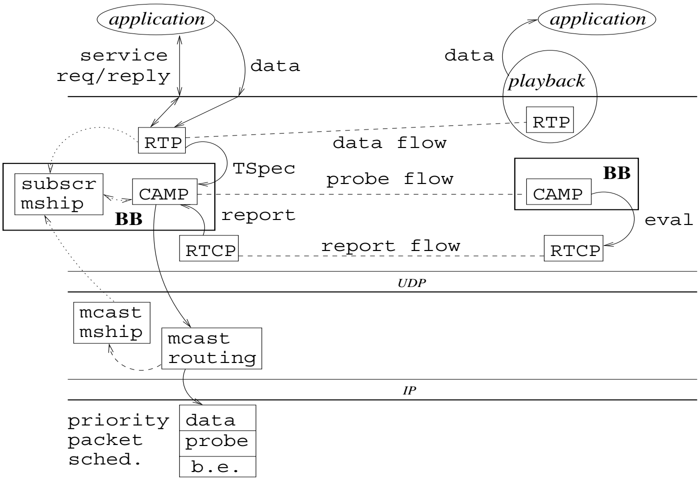


Fig. 1. Architecture of Camp-based end stations

the aim of evaluating whether the available bandwidth is sufficient to support the new traffic. With respect to the classification given in [2], we adopt *out-of-band* probing with *dropping* of the `probe` packets as the congestion signal. All the routers use a *priority* packet scheduling discipline: the `probe` packets are marked with a higher priority than the best effort packets, but a lower priority than the QoS packets. This priority assignement ensures that the probing traffic does not affect the established QoS sessions. On the other hand, `probe` packets can drain the available bandwidth for the new QoS session at the expenses of the best effort traffic.

To support multicast, two issues must be considered: (*i*) the receiver group membership can dynamically change; and (*ii*) different destinations can experiment different QoS in receiving the same traffic. To cope with problem (*ii*) above, the recipients that receive the probing traffic with low quality prune from the tree and refuse the service, by sending a refusing RTCP report to the source. When all the reports have been received by the source, if the service is accepted by at least one recipient, the source switches from the transmission of `probe` packets to the transmission of the data packets generated by the application, *without discontinuity*. The data packets are forwarded along the pruned tree.

We deal with the problem (*i*) above using a *proxy* mechanism. The source announces the multicast session via `sdr` and starts transmitting `probe` packets at the scheduled time, if at least one receiver is listening. A Camp proxy is instantiated in a router either in the initialization phase, or when one or more new downstream output interfaces (*oif*s) appear in the router for the group (dynamic membership changes). The proxy remarks as `probe` packets all the incoming packets for the session, that must be forwarded to the probing *oif*s. This way, the data sent to the new destinations do not affect previously established sessions traversing the new branch. The proxy lifetime lasts until, for each probing *oif*, either it is pruned from the tree (as the result of a service refusal), or an acceptance report is received from it. In the latter case, if the initialization phase is ongoing, the report is forwarded to the source. The source Camp entity switches to the transmission of the data as soon as it receives an accepting report. The proxy mechanism allows to hide the membership to the source.

3 Performance Evaluation

We have implemented the architecture shown in Figure 1, in the frame of the ns-2 simulation package [3]. The simulations have been performed with a meshed network of 64 nodes, connected by optical links of 2 Mbps bandwidth and variable length in the range 50 to 100 Km. Background, best effort traffic is uniformly distributed all over the network; best effort sources generate CBR traffic with a 0.66 Mbps rate. The size of best effort, `probe` and data packets is 512 bytes. We embedded a real RTP implementation into the RTP template of ns-2. The recipients dynamically join the group; we performed experiments with different join rates. The multicast tree is incrementally built as join events occur; the source is located in the tree root. The source does not know the group of recipients; it generates CBR traffic whose rate assumes different values in the range 0.4 Mbps to 1.9 Mbps. During the probing phase, the recipients compare the received rate with the source rate: if the difference is below a tolerated threshold, a recipient

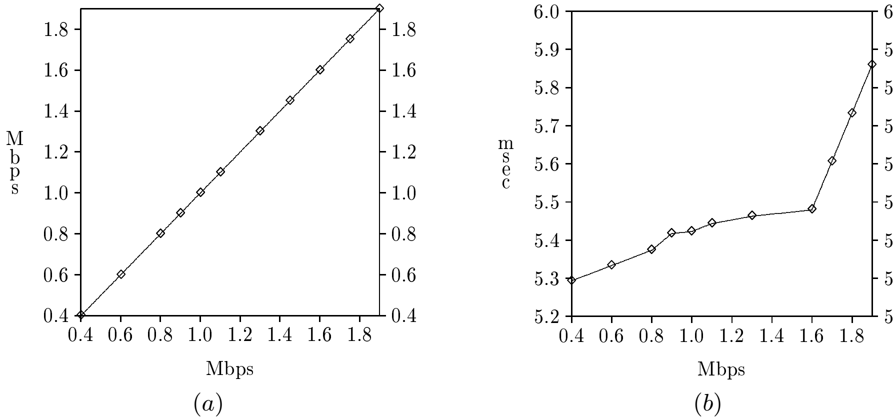


Fig. 2. (*a*) Throughput vs. offered load for $|G| = 10$. (*b*) Average end-to-end delay vs. offered load for $|G| = 10$

sends a positive report. We performed measures for different thresholds [5]. The recipient decision is sent within the first RTCP report a destination generates after the reception of a number of `probe` packets, i.e. of samples, sufficient to ensure an accurate measure of the available bandwidth by covering the rate of the slowest traffic source. We performed simulations with different sample sizes.

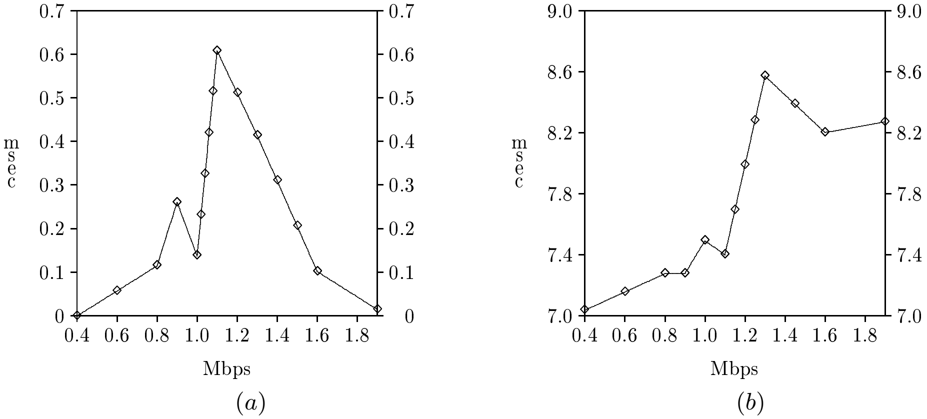


Fig. 3. (*a*) Jitter vs. offered load, for $|G| = 10$. (*b*) Fair delay vs. offered load, for $|G| = 10$

By performing simulations with different group cardinalities, we observed that the performance is almost independent of this parameter. We performed simulations with recipients that join the group with different rates. The proxy mechanism has proved to be effective in performing the admission control, and the measured performance is independent of the frequency with which recipients join the group. The results shown in this section have been obtained for a group of 10 recipients, acceptance threshold set to 5% of the source rate and frequency of the join requests arrival 1 sec. The measures have been taken after 20 sec. from the end of the set-up phase of the last grafted recipient.

Simulations indicate that CAMP effectively performs the call admission control. The recipients accepting the transmissions receive at the correct data rate. In Figure 2, we report the throughput (*a*) and the end-to-end delay (*b*) averaged over all the recipients; no receiver has refused the service. The delay increases when the offered load approximates the link capacity, while it is independent of the interference of the best effort traffic. This indicates that the sessions characteristics are preserved from source to destination, independently of the other network load.

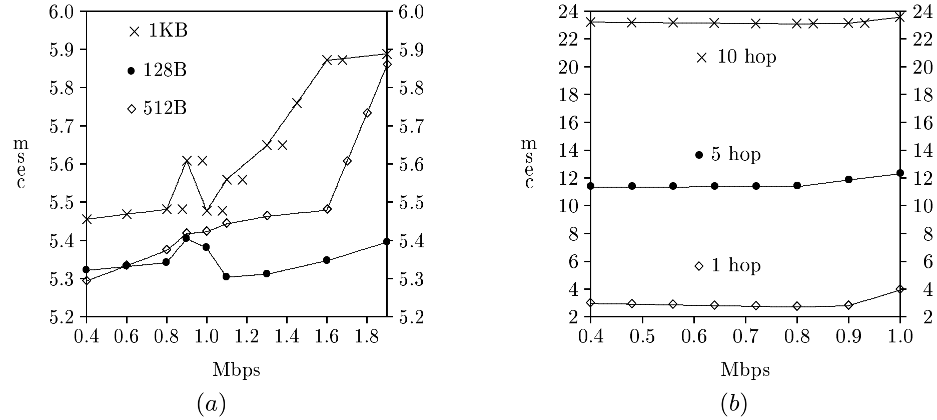


Fig. 4. (*a*) Average end-to-end delay vs. offered load as a function of the best effort packet size, for $|G| = 10$. (*b*) Average end-to-end delay vs. offered load as a function of the receivers distance from the source

The jitter has been computed according to the algorithm given in the RTP specification [6]; it is reported in Figure 3(*a*). The jitter behaviour indicates that at the receiver side a delivery agent must be used to perform the playback of the source transmission. The jitter shows a peak in correspondence with the maximum contention between the QoS and the best effort traffic. After that value, the QoS traffic pushes the best effort traffic away from the tree branches, and best effort packets start to be dropped from the queues.

The *fair delay* is the maximum difference between the end-to-end delays perceived by two different destinations. Its behaviour (Figure 3(*b*)) indicates that the destinations at a greater distance from the source greatly suffer the network congestion. In the worst case, this could result in a lower probability of successful service set-up for the farest destinations. However, we never observed service refusal.

The measures of the jitter and the fair delay show the effects of the presence of best effort packets at the core routers. As expected, the priority mechanism alone is not sufficient to ensure jitter control at the destinations. To highlight the impact of the best effort traffic over the QoS, we performed simulations with different best effort packet sizes. In Figure 4(a) we report the end-to-end delay observed by the QoS packets that compete with best effort traffic generated as before. As the links cannot be preempted once a packet transmission is ongoing, QoS packets arriving at a node could have to wait at most for a best effort packet transmission time before gaining the link, although they have the highest priority. The impact of the delay over the received rate is however negligible.

We performed an experiment with two sources: the former one has a 1 Mbps rate; we varied the rate of the latter source. In figure 4(b), we show the average delay measured with respect to the load generated by the second source and the distance of the recipients from the source. The contention probability amongst different sessions increases with the path length: it affects the queueing delays, thus altering the regular traffic profile. The impact on the received throughput is however negligible.

The achieved results show that the devised mechanism effectively performs admission control. Yet, further investigation has to be carried out concerning the interactions amongst several concurrent transmissions and their impact on the probability of successful service establishment.

References

1. Blake S., Black D., Carlson M., Davies E., Wang Z., Weiss W. "An Architecture for Differentiated Services". *RFC 2475*, Dec. 1998. Work in progress.
2. Breslau L., Knightly E., Shenker S., Stoica I., Zhang H. "Endpoint Admission Control: Architectural Issues and Performance". *Proc. SIGCOMM'00*, 2000, pp. 57-69.
3. Fall K., Varadhan K. "ns Notes and Documentation". The VINT Project, Jul. 1999, `http://www-mash.CS.Berkeley.EDU/ns/` .
4. Nichols K., Jacobson V., Zhang L. "A Two-bit Differentiated Services Architecture for the Internet". *Internet Draft, draft-nichols-diff-svc-arch-00.txt*, Nov. 1997. Work in progress.
5. Pagani E., Rossi G.P., Maggiorini D. "A Multicast Transport Service with Bandwidth Guarantees for Diff-Serv Networks". *Lecture Notes in Computer Science 1989*, Jan. 2001, pp. 129. Springer, Berlin.
6. Schulzrinne H., Casner S., Frederick R., Jacobson V. "RTP: A Transport Protocol for Real-Time Applications". *RFC 1889*, Jan. 1996. Work in progress.
7. Teitelbaum B., Chimento P. "QBone Bandwidth Broker Architecture". *Internet 2 QoS Working Group Draft*, Jun. 2000. Work in progress.
   `http://qbone.internet2.edu/bb/`

A Framework to Service and Network Resource Management in Composite Radio Environments[1]

L.-M. Papadopoulou, V. Stavroulaki, P. Demestichas, and M. Theologou

National Technical University of Athens (NTUA),
Electrical and Computer Engineering Department, Telecommunications Laboratory,
9 Heroon Polytechneiou Str, 15773 Zographou, Athens, Greece
louisa@telecom.ntua.gr

Abstract. This paper builds on the assumption that in the future, UMTS, HIPERLAN-2 and DVB-T can be three (co-operating) components of a composite radio infrastructure that offers wideband wireless access to broadband IP-based services. Managing the resources of this powerful, composite-radio infrastructure in an aggregate manner, and multi-operator scenario, is a complex task. This paper presents an approach to the overall UMTS, HIPERLAN-2 and DVB-T network and service management problem, providing the internal operation of a system addressing this problem. Key points addressed are the development of an architecture that can jointly optimise the resources of the technologies in the composite radio environment, and the development of open interfaces with Service Provider mechanisms and the heterogeneous managed infrastructure.

1 Introduction

Wireless systems continue to attract immense research and development effort [1], especially in the following areas. First, the gradual introduction of third generation systems like the Universal Mobile Telecommunications System (UMTS) [2] and the development of the IMT-2000 framework [3]. Second, the standardisation, development and introduction of Fixed Wireless Access (FWA) systems, which support radio access to broadband services, with limited mobility; a pertinent promising example is the HIPERLAN (High Performance LAN) initiative [1]. Third, the advent of Digital Broadcasting Systems (DBS) like the Digital Video Broadcasting (DVB) and the Digital Audio Broadcasting (DAB) initiatives [4]. Moreover, a recent trend (compliant with the features envisaged for the Fourth Generation (4G) wireless systems' era) is to assume that UMTS, HIPERLAN-2 and DVB-T will be three co-operating wireless access components.

In other words, UMTS, HIPERLAN-2, and DVB-T can be seen as parts of a powerful, *composite-radio*, infrastructure through which their operators will be

[1] Work partially funded by the CEC, under the 5th Framework Program, within the IST project MONASIDRE (IST-2000-26144: Management of Networks and Services in a Diversified Radio Environment).

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 1190-1195, 2002.

enabled to provide users and service providers (SPs) with alternatives regarding the efficient (in terms of cost and QoS) wireless access to broadband IP-based services. This paper presents the development of a *UMTS, HIPERLAN-2 and DVB-T network and service management system* capable of:

- Monitoring and analysing the statistical performance and QoS levels provided by the network elements (segments) of the managed infrastructure, and the associated requirements originating from the service area (environment conditions, e.g., traffic load, mobility levels, etc.).
- Inter-working with SP mechanisms, so as to allow SPs to dynamically request the reservation (release, etc.) of network resources.
- Performing dynamic reconfigurations of the overall managed UMTS, HIPERLAN-2 and DVB-T infrastructure, as a result of resource management strategies, for handling new environment conditions and SP requests in a cost-efficient manner.

In the following, the management architecture in the aspect of a composite radio and multi-operator context, and the operation of such a system are presented.

2 Management Architecture in a Composite Radio and Multi-operator Context

Our model of the composite radio environment includes three different wireless access technologies and has a flexible implementation. In the context of this paper, each wireless access system is considered to belong to a different operator, occupying a network and service management platform. A generic management architecture of such a platform is split in three logical entities as follows.

- *MASPI* (Monitoring and Assessment and SP mechanism Interworking). This component captures the (changing with time) conditions that originate from the environment (service area) of the managed UMTS, HIPERLAN-2 and DVB-T infrastructure; this is accomplished by monitoring and assessing the relevant network and service level performance of the managed network elements and segments. This component also interworks with the SP mechanisms, so as to allow SPs to request the reservation of resources (establishment of virtual networks) over the managed network infrastructure. Virtual networks are seen as the realisation of contracts that the managed system should maintain with SPs.
- *RMS* (Resource Management Strategies). This component applies resource management strategies, so as to dynamically find and impose the appropriate UMTS, HIPERLAN-2 and DVB-T infrastructure reconfigurations, through which the service provider requests, and/or the (new) service area conditions, will be handled in the most cost-efficient manner.
- *NES* (UMTS, HIPERLAN-2 and DVB-T Network and Environment Simulator). It provides the means for validating some management decisions prior to their application in the real network. This component enables off line testing, validation and demonstration of the management mechanisms.

The management system components are distributed in each domain, specialised for handling the specific (UMTS, HIPERLAN-2 and DVB-T) technology. However, these components can co-operate for handling SP requests and/or new environment conditions.

3 System Operation

A sample scenario according to which the components above collaborate is provided in Fig. 1. MASPI-U, RMS-U, and NES-U represent components dedicated to the UMTS network (similarly for the HIPERLAN-2 and DVB-T networks). The interactions with the NES-U, NES-H and NES-D components are omitted for simplicity reasons.

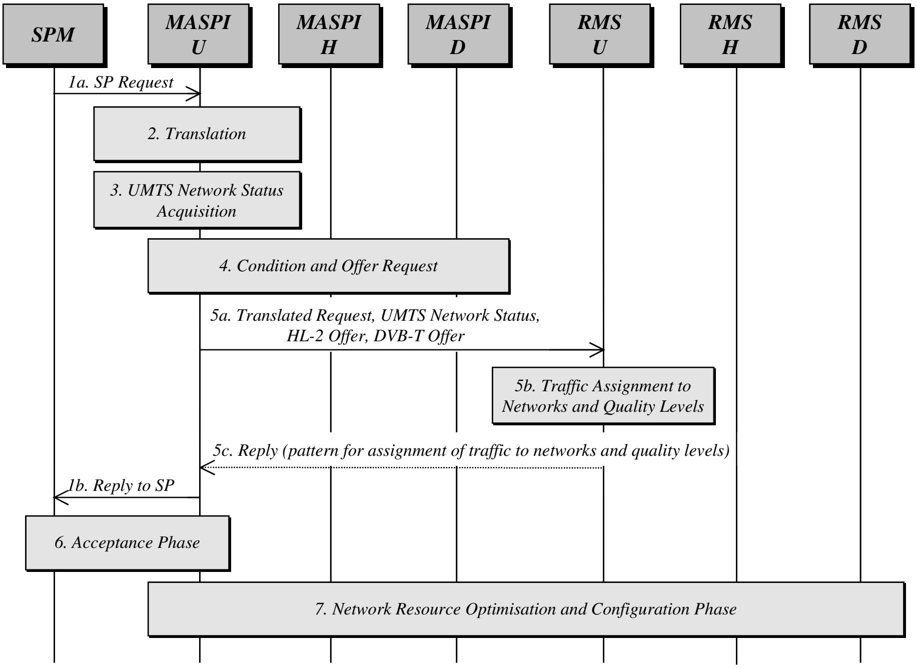

SPM
MASPI U
MASPI H
MASPI D
RMS U
RMS H
RMS D
1a. SP Request
2. Translation
3. UMTS Network Status Acquisition
4. Condition and Offer Request
5a. Translated Request, UMTS Network Status, HL-2 Offer, DVB-T Offer
5b. Traffic Assignment to Networks and Quality Levels
5c. Reply (pattern for assignment of traffic to networks and quality levels)
1b. Reply to SP
6. Acceptance Phase
7. Network Resource Optimisation and Configuration Phase

Fig. 1. Sample operation scenario. The scenario shows how the components of the UMTS, HIPERLAN-2 and DVB-T network and service management system collaborate

The initiation of the scenario is done from the UMTS network, as an example. The procedure would be similar if the MASPI component observed a new environment condition (e.g., alteration in the traffic demand, mobility and interference levels, etc.) in the managed network. Alternatively, the process could have been initiated by a SP request towards the HIPERLAN-2 or DVB-T management system. The network that receives a SP request or observes a new environment condition is called originating network.

The scenario consists of steps that can be roughly categorised in four phases. In the first phase (step 1), the SP issues a virtual network establishment request. In the second phase (steps 2-5), the request is processed by the UMTS, HIPERLAN-2 and DVB-T network and service management system (this includes translation of the SP request from a service to a network level view, network status acquisition of the originating network, condition and offers from the co-operating networks, and traffic assignment to networks and quality levels). In the third phase the proposed solution is accepted by the parties involved (SP and network operators, step 6). Finally, in the fourth phase the managed systems are appropriately configured (step 7). The aforementioned steps are addressed in detail in the following sub-sections.

3.1 Service Provider Request

MASPI supports the functionality of handling the SP requests and the corresponding replies to these requests, as a general framework of the processing, establishment and maintenance of contracts (SLAs) with the SPs. A typical SP request has the following general structure (content). (i) It specifies the service (or set of services), the provision of which the SP requests from the management system. (ii) It specifies the distinguishable user classes to which the service is offered; each user class is associated with specific quality levels, a user class profile and a terminal profile. The quality of service levels express the quality levels that are considered acceptable for the provision of services to users (subscribers) that belong to the specific user class. If the quality levels are more than one, the SP may also provide the significance factor of each quality level for the present user class. The user class profile includes mobility, and traffic characteristics of the users, and is described in a file with specific format. The terminal profile assists the management system on knowing which networks can be used to satisfy the SP request (e.g., users of a specific user class may have terminals that support only the UMTS network). (iii) It specifies the number of subscribers that correspond to each service and user class. (iv) It includes information about the area (geographic region) to which the request is applied to, and the time zone, i.e., the time period during the day that the service should be provided to the users of the specific user class.

3.2 Service Provider Request Translation

MASPI supports the functionality of translating the SP request from a service level view to a network level view. The information about the requested services, and user classes (including user class profiles and quality of service levels) is used in order to investigate various options regarding the network load required to fulfill the request. On the other hand, the area information in the SP request can be exploited for the detection of the cells that will be affected by the SP request.

3.3 Network Status Acquisition

During this task the status of the originating network (e.g., traffic carried by cells that can be affected by the SP request) is obtained. MASPI maintains interfaces with the underlying network interface and/or the network element management infrastructure. MASPI collects service level information based on the handled load, the provided performance (measured e.g., by the blocking probability, the dropping probability or the delay), and the dedicated resources per service provider, service, and user class. An integration of this service level information about the management system enables a network level provisioning of the managed system infrastructure. In case of performance degradations MASPI can initiate the procedure of the scenario described in Fig. 1.

3.4 Condition and Offer Request

Each MASPI is capable of requesting from the co-operating networks' MASPIs the amount of resources or the load that these networks can carry, as well as cost related information. Likewise, each MASPI is in position to respond to such requests. This information (bandwidth availability and cost), as well as other information (e.g., the area and time zone for which the services should be provided), will contribute to the decision of the traffic splitting between the three networks.

3.5 Traffic Assignment to Networks and Quality Levels

This is an optimisation problem, targeted to the splitting of the traffic to the three networks and the assignment to quality levels. Considering the case of the scenario depicted in Fig. 1, this optimisation problem relies on the following input data:

- The translated SP request, which can express the service demand per user class;
- The benefit deriving from the assignment of (portions of the) service demand to the several quality levels;
- The status of the UMTS, HIPERLAN-2 and DVB-T network segments that are to be affected by the request;
- The UMTS, HIPERLAN-2 and DVB-T offers, i.e., the cost that these networks will impose per quality level of the service.

The optimisation results to an allocation of the service demand to networks and quality levels. The allocation should optimise an objective function, which is associated with the amount of the service demand accommodated, the quality levels at which the service demand will be accommodated, and the benefit deriving from the assignment of service demand to high quality levels.

The constraints of the optimisation problem fall into the following categories:

- The service demand should be assigned to acceptable quality levels;
- The capacity constraints (deriving form the UMTS network status, and the HIPERLAN-2 and DVB-T condition and offers) should not be violated.

3.6 Reply to Service Provider Request – Acceptance Phase

The MASPI component reply to the initial SP request includes the quality levels to which the user classes are assigned, cost related information per user class, as well as the volume of the subscribers assigned to each network. This information is valid for the area and time zone specified in the SP request. The acceptance of this reply from the SP will lead to the establishment of a SLA.

3.7 Network Resource Optimisation and Configuration

After the decision by the RMS on the traffic allocation to the three networks, and the acceptance phase, the MASPI of the originating network makes a resource reservation request to the other networks' MASPIs, in order these systems to accommodate the assigned traffic. Thereafter, optimisation and configuration procedures take place among the three networks. RMS finds an optimal configuration of the managed network segments, so as to guarantee that the traffic assigned to them will be handled (carried) with the most cost-efficient manner. This part of the management system consists in a suite of tools and procedures that optimise functions including for instance cost, network performance criteria, etc., under a set of constraints related to target QoS levels, resource utilisation, fault tolerance, etc.

4 Conclusions

This paper builds on the assumption that in the future, UMTS, HIPERLAN-2 and DVB-T can be three (co-operating) components of a composite radio infrastructure that offers wideband wireless access to broadband IP-based services. In this direction the paper presented an approach to the overall UMTS, HIPERLAN-2 and DVB-T network and service management problem, addressing the operation of a system that deals with such a problem.

The paper, or alternate versions of this paper, can be expanded with more information on the internal functionality of the components, the provision of details regarding the design choices followed, or the presentation of indicative results obtained from case studies. The application of the management architecture in large-scale network test-beds is a future stage of our work.

References

1. U. Varshney, R. Vetter, "Emerging mobile and broadband wireless networks", *Commun. of the ACM*, Vol. 43, No. 6, June 2000
2. "Wideband CDMA", Feature topic in *IEEE Commun. Mag.*, Vol. 36, No. 9, Sept. 1998
3. "IMT-2000: Standards effort of the ITU", Special issue on *IEEE Personal Commun.*, Vol. 4, No. 4, Aug. 1997
4. Digital Video Broadcasting Web site, www.dvb.org, Jan. 2001

JESA Service Discovery Protocol
Efficient Service Discovery in Ad-Hoc Networks

Stephan Preuß

University of Rostock; Dept. of Computer Science;
Chair for Information and Communication Services*
`mailto:spr@informatik.uni-rostock.de`
`http://wwwiuk.informatik.uni-rostock.de/staff/spr.html`

Abstract. Pervasive computing requires management techniques allowing efficient service handling in volatile contexts. The Java Enhanced Service Architecture (JESA) is a service platform addressing this issue for resource limited devices. A major problem in dynamic service networks is the discovery of desired services. The *JESA Service Discovery Protocol (JSDP)*, one of JESA's core components, is a lightweight, platform independent service discovery protocol for ad-hoc networks. JSDP features transparent operation with or without central service brokers providing scalability from point-to-point connections to larger structured networks.

Keywords: service discovery, pervasive computing, ad-hoc networking

Classification (CR 1998): C.2.2, C.2.3, C.2.4

1 Introduction and Motivation

In ad-hoc networking, client nodes enter an initially unknown territory for using network services. During the *service discovery* process, they gather information about their surrounding service context avoiding the inflexible use of only well known or preconfigured services. A discovery technology meeting a broad range of application areas is characterized by: *scalability concerning service count and resource usage*, *platform independence concerning computing and network*, *low complexity for easy application development*, and *application transparency*. The existing discovery technologies (e.g. SLP [1], Ninja SDS [2], SSDP [3], Jini [4], Salutation [5]) do not meet all of the above characteristics.

Especially, the need for a service platform applicable to resource limited embedded or mobile devices as well as to desktop and server systems led to the development of the *Java Enhanced Service Architecture (JESA)* [6] – a lightweight, Java-based middleware for spontaneous service discovery and usage. It is mainly intended to be used in industry and home automation as well as mobile computing. One of its core components is the *JESA Service Discovery Protocol (JSDP)* – an efficient discovery mechanism offering operation modes of different complexity to be applicable to a wide variety of device classes.

* Supported by a grant of the Heinz Nixdorf Stiftung.

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 1196–1201, 2002.

2 Service Discovery

The establishment of service relations in volatile network environments requires the nodes to become aware of their service context and context changes. *The information necessary for using a specific service is gathered in the service discovery process.* Service discovery comprises at least one of the following items: *locating the service provider*, *acquiring additional service or provider information*, *retrieving the provider's access interface (proxy, stub, etc.).*

Service discovery can be classified according to the level of initial knowledge, the relation, the count, and the activity of the involved entities. The distinct discovery classes build a hierarchical structure as laid out in Fig. 1. Service discovery is split up into two top-level categories. *Preconfigured* discovering entities know either about the desired service provider or whom to ask for that information. In contrast, *non-configured* entities are innocent regarding the service context. This is the typical situation for ad-hoc networks. Further subdivision relates to the number of involved entities. *Location-aware* and *immediate* discovery classes are characterized by direct relation between client and provider. The provider itself supplies the client with all necessary information. In *mediated* modes, service brokers deal discovery information on behalf of providers. At the bottom-level, five discovery categories are distinguished: *Location-aware* clients know the logical location of their desired service, hence discovery reduces to getting additional information like service access interface or service characteristics; In *active* mode, a client initiates a request response procedure by broadcasting a request for a certain service. Appropriate providers respond at least with the location data of their service; *Passive* mode, in contrast, releases clients from inquiry and obliges providers to announce their services; Both *mediated* modes work with central information brokers. For proper operation providers have to register their service data with a broker. For that purpose, a broker may be treated as a special service provider. Regular providers will be clients of the broker for the registration period and have to discover the broker service by any of the discovery means discussed here. *Transparent* and *non-transparent* mode differ in the client's awareness of the broker. If a client intentionally used a broker for finding services it discovers *non-transparently.* Whereas, if the client believed to interact directly with a provider but in fact it was a broker, the client discovers *transparently.*

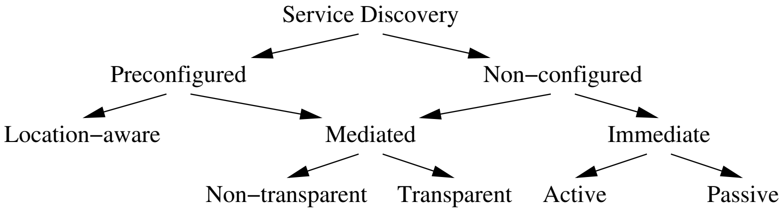


Fig. 1. Service Discovery Categories.

3 Related Technologies

There is a number of ad-hoc service platforms deploying different discovery schemes: *SLP* offers a comprehensive message based discovery system for TCP/IP networks; *SSDP* is the discovery protocol used in *UPnP* [7], it mainly bases on UDP, HTTP, and XML; The discovery protocol of the Ninja project, *SDS*, enables secure service discovery with an infrastructure of service, accounting, and certificate directories; *Salutation* provides generic network independent service discovery and access by the use of network and service managers; *Jini* features Java-based, broker-centric discovery and service usage with proxies.

Table 1. Discovery Classes Supported by Current Protocols.

	Location-aware	Non-transparent	Transparent	Active	Passive
SLP		+		+	+
SSDP	+			+	+
SDS		+			
Salutation		+		+	+
Jini		+			
JSDP		(+)	+	+	+

Table 1 presents an overview of the protocols' discovery class conformance, according to the classification given in Section 2. Furthermore, it contains a forecast of the JSDP functionality.

4 JESA Service Discovery Protocol

JSDP has been designed as a lightweight discovery protocol for embedded and mobile systems. It provides generic functionality: *locating of service providers, retrieval of service proxies and service attributes.* Almost any discovery feature can be added leveraging service attributes. Integral security mechanisms have been left off the protocol to keep it small. A major goal is JSDP's ability to work transparently *with or without* a central service broker (see Section 2) thus offering the possibility of peer-to-peer communities of limited devices on the one side and large scalable service networks on the other side.

4.1 Discovery Strategies

JSDP incorporates three major discovery modes (see Fig. 2). The *immediate* ones (*active, passive*) are intended for short range discovery in the local network segment. The *transparent* mode enables long distance discovery across segment boundaries by using a broker hierarchy. Brokers may either forward service requests or share the registered service information. With the existing means, *non-transparent* discovery can be realized by special broker services having not

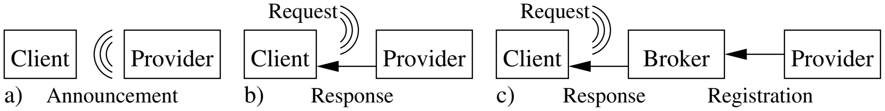


Fig. 2. JSDP Operation Modes: a) passive; b) active; c) transparent.

only a registration interface but a query interface, too. This architecture would be similar to Jini.

Service discovery is performed in two steps. In the first step, provider location information is gathered using the *Provider Location Protocol (PLP)*. If a client decides to further examine or to use a certain service, in the second step, proxy and/or attribute information will be retrieved by the *Proxy/Attribute Request Protocol (PARP)*.

4.2 Provider Location Protocol

The *Provider Location Protocol (PLP)* can be used in request response mode (active discovery) or in announcement mode (passive discovery) for acquiring service provider locations. For simplicity, it defines only a single message format for requests and announcements (see Fig. 3). Service queries or announcements contain only the service *type* because this is the most important selection criterion. If a client needs more information about a service it will continue with the attribute retrieval and come to a decision according to the service characteristics. The *group list* can be used to limit the service matches to certain administrative groups. A query will match a service if version and type are identical and the service is member of at least one listed group. The *ID list* can contain a set of *Universal Unique Identifiers (UUID)* which specify services a client does not want to get announced. A provider fills in this list a single UUID which is used by clients for further transactions. Although PLP is designed in a Java environment, it is not Java-specific. Hence, it can be used for service discovery in non-Java environments as well. The current PLP implementation bases on UDP. Requests or unsolicited announcements are delivered with UDP multicasts; alternatively, link-local broadcasts can be used. Solicited announcements (service responses) are sent by UDP unicasts. The message flow for active discovery accords to Fig. 2b. A client uses the *sender URI* field to tell potential responders where to send the answer. The sender URI field in the answer codes the continuation point for the next discovery steps, in fact where to contact the provider to get the attributes or the service access interface.

Version	Sender URI	Type	ID Count	ID List	Group Count	Group List
Int32	UTF8 String	UTF8 String	Int32	[Int128]	Int32	[UTF8 String]

Fig. 3. Service Request / Announcement Message.

Transparent discovery mode uses the same message exchange procedure. From the client's point of view nothing changes. Providers have to register their service data (location information and access interface) with a broker and stop answering requests or announcing services. Brokers should be implemented as regular JESA services discoverable by JSDP means. Now, the broker answers requests for registered services and delivers their proxies or attributes on request. The unsolicited announcement of registered services (passive discovery) is possible but not encouraged.

4.3 Proxy/Attribute Request Protocol

After successful completion of PLP, a client knows about at least one provider's or broker's location and can fetch a service proxy or service attributes using the *Proxy/Attribute Request Protocol (PARP).* In contrast to PLP, PARP is more dedicated to a Java environment because it ships serialized Java objects (JESA service proxies and attributes [8,6]) across the network. PARP uses a stream connection, currently it is TCP, to request and transmit service data. The Request message (see Fig. 4) indicates in the tag field whether proxy or attribute list is to be transmitted. The service UUID field exactly identifies the target service and has been retrieved in a PLP response. In immediate modes, directly talking to a provider, the UUID would be almost superfluous. It is possible that in the time between PLP and PARP execution a certain provider leaves the community and a new one enters with the same logical location. Here, the UUID avoids the delivery of incorrect service data. In transparent mode, using a broker, the UUID is used to select the appropriate information out of the pool of registered service data. If a service matching the UUID is available service data, according to the tag field, will be returned to the client in form of serialized Java-objects.

Tag	Service UUID
int32	Int128

Fig. 4. PARP Request.

For robustness, transparent operation mode can be split up. During PLP, the broker does not deliver its own URI for further discovery but the one of the original provider. Hence, proxy or attributes will be fetched from the real provider. Successful completion of this step ensures a properly working provider.

Clients should not cache service data over long periods and reuse them because there is no guarantee that a service remains available and does not change its logical location or characteristics. For most cases a "discover-use-forget" strategy will be appropriate.

5 Conclusion and Future Work

JSDP has been developed as integral discovery component of the Java Enhanced Service Architecture. It focuses on core discovery tasks and allows arbitrary extensions. By the seamless integration of immediate and mediated discovery modes JSDP scales with the service count. The current implementation is fully functional on top of TCP/IP networks. JSDP itself comes with a memory footprint of about 30K. Together with the NetObjects technology [8] for service execution, a properly working JESA system requires about 80K. JESA and hence JSDP require a Java1.1 compliant runtime environment and will fit to embedded or mobile systems running PersonalJava, Kaffe, Jeode, J2MECDC.

Ongoing developments will produce: 1) *Java Abstract Network (JANet)* which decouples JESA from TCP/IP, a reference implementation will run with the CAN-bus; 2) An OSGi [9] interface which automatically transforms OSGi services into JESA services discoverable by JSDP; 3) An integration of the Bluetooth SDP [10] into JSDP to avoid multiple discovery levels when applying JESA to Bluetooth devices.

References

1. Guttmann, E., Perkins, C., Veizades, J., Day, M.: Service Location Protocol, Version 2. IETF Internet Draft, RFC 2608 (1999)
2. Czerwinski, S.E., Zhao, B.Y., Hodes, T.D., Joseph, A.D., Katz, R.H.: An Architecture for a Secure Service Discovery Service. In: Proceedings of the Mobicom 99, Seattle, Washington, USA, ACM (1999) 24–35
3. Goland, Y.Y., Cai, T., Leach, P., Gu, Y., Albright, S.: Simple Service Discovery Protocol/1.0. `http://www.upnp.org/download/draft_cai_ssdp_v1_03.txt` (1999)
4. Arnold, K., Wollrath, A., O'Sullivan, B., Scheifler, R., Waldo, J.: The Jini Specification. Addison-Wesley (1999)
5. Salutation Consortium Inc.: Salutation Architecture Specification V2.1. `ftp://ftp.salutation.org/salute/s21a1a21.pdf` (1998)
6. Preuß, S.: Java Enhanced Service Architecture. `http://wwwiuk.informatik.uni-rostock.de/~spr/jesa/` (2001)
7. UPnP Forum: Universal Plug and Play Connects Smart Devices. `http://www.upnp.org/` (1999)
8. Preuß, S.: NetObjects - Dynamische Proxy-Architektur für Jini. In: Proceedings of Net.ObjectDays 2000, Erfurt, c/o tranSIT GmbH (2000) 146–155
9. The Open Services Gateway Initiative: OSGi Service Gateway Specification. `http://www.osgi.org/` (2000)
10. SIG, B.: Core. In: Specification of the Bluetooth System. Volume 1. Bluetooth SIG (February 2001)

Performance Simulations of a QoS Aware Caching Method

Pertti Raatikainen [1], Mika Wikström [2], and Timo Hämäläinen[2]

[1] VTT Information Technology, Telecommunications
P.O.Box 1202, FIN-02044 VTT, Finland
pertti.raatikainen@vtt.fi
[2] University of Jyväskylä, Department of Mathematical Information Technology
P.O. Box 35, FIN-40351 Jyväskylä, Finland
timoh@cc.jyu.fi, wikstrom@mit.jyu.fi

Abstract. Research of web-servers has recently addressed the problem of content distribution coupled with quality of service (QoS). Due to the explosive growth of services offered over the Internet, novel mechanisms are needed for IP based service delivery to scale in a client-transparent way. This paper addresses the above problem considering also utilization of available processing power of servers. Many developed caching systems dedicate a fixed portion of the processing power for higher QoS services leading to lowered overall throughput of the server system. Here we introduce and simulate a QoS aware caching scheme that offers lower response delay for higher quality services and additionally optimizes utilization of the available processing power.

1. Introduction

Distribution of web-content to servers and arbitration of requests among a cluster of servers have been in focus of intense research. Location of content among the servers and service admission control are the major problems in server farm implementations. Since the same content can be located to several servers, an additional problem appears in maintaining an established connection to specific content on a known server throughout a session. A number of different schemes to locate content to servers and balance loading between them have been developed [1, 2, 3, 4, 5]. The most sophisticated ones, often called web-switches, are content aware and offer methods to maintain connection to a given server all along a session [10].

The content based switching schemes enable categorization of connections, e.g., based on the requested service, user or combination of both. Good customers or access to certain high quality services may be directed to less loaded servers enabling lower response delay. This causes skewing of the processing balance and at worst some servers are overloaded while others remain lightly loaded. Optimum loading implies that loading degree of each server can be fixed to be equal.

Caching combined with content aware switching is a technique that can be used in lowering response delay for some customers or services, while maintaining the loading balance between the servers. Since the number and size of web-files is normally large compared to the available cache size, a subset of web-pages can be located to cache

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 1202-1207, 2002.

[5, 7, 8]. To maximize the number of cache hits when the cache capacity is exceeded, a number of caching algorithms have been developed to overwrite less frequently accessed pages with more frequently accessed ones [3, 6, 7].

This paper introduces a caching method that allows to distribute web-content based on QoS requirements and simultaneously balance load among a cluster of servers to enable maximum utilization of the aggregate processing power. The objective in locating web-files to cache is to maximize cache hit-rate and thereby minimize response delay. Chapter 2 introduces the caching algorithm and related simulation model. Chapter 3 gives some simulation results and chapter 4 concludes the paper.

2. QoS Aware Caching Scheme

The objective in developing the caching scheme was to improve cache system performance measured in service response delay and utilization of the server system's processing power. Response delay is lowered by increasing the cache hit-rate and utilization of processing power is optimized by locating content randomly to servers. When the number of web-files is large, random location policy leads to uniformly distributed load among the servers.

2.1 Caching Method

Each server in a cluster is supposed to have a hard disk and cache memory of known size. The memory sizes, processing power and memory reading delays may vary from a server to another. Web-files stored on servers are categorized into a fixed number of QoS classes. In a general case, the number of files in those classes and sizes of the files are different.

The web-files are located randomly to servers and upon storing a file it is associated with one of the QoS classes. If cache is small compared to the aggregate size of files in a server system, a predetermined percentage (p_c) of files from each QoS class can be located to cache. The highest QoS classes have priority over the lower classes and the most requested files of each QoS class are selected first. Files that cannot be stored in cache are located on hard disk. If the cache is large enough to store a fixed percentage p_c of files from all QoS classes then some of the leftovers can also be placed to cache. The order in which they are stored follows the QoS class priorities and request rate intensities of the files.

2.2 Simulation Model

The simulations are carried out by applying a generic cache simulation model introduce in [9]. It allows manipulation of a number of parameters that characterize the introduced QoS based caching scheme. Performance of the system can be studied by varying the cache sizes, number and size of web-files, number of QoS classes, number of files in each QoS class, file request rates and processing power of servers.

Logically the model is divided into five decision-making blocks (see Fig. 1). At the top is a block, which decides whether the next event is related to an incoming or outgoing request. A request coming from a client is considered an incoming one and a request that has been processed and is being directed to a server an outgoing one.

In case of an incoming request, the algorithm first chooses the server that has the requested file. Different sorts of selection algorithms can be implemented based on the selected file location policy. Then the algorithm checks whether the selected server is available for service. If it is not available, the request is put to into a queue where it will stay until the algorithm enters the *outgoing* leg.

If the server is available, the simulation enters the block, which checks whether the requested file is in the cache. Different policies can be used in placing files to the cache. Here, priority is given to files of the highest QoS classes. If the requested file is in the cache, it is read and marked as the most recently accessed one. If the file is not found in the cache, it must be on the hard disk and the simulation continues to the „*Read file*" -block. The file is read from the disk and the deployed caching scheme decides how to proceed (store it to the cache or leave on the disk). This block allows the use of different caching methods and comparison of their performances.

If in the topmost block the *outgoing* leg is chosen then the server where the file is being served is located and the file is removed from service. After that, it is checked whether there is a queue for that particular service. If there is no queue, the model starts another iteration round, i.e., it starts to process the next event. If there is a queue for that service then the first request in the queue is serviced.

Furthermore, the server system keeps record of all file, their QoS classes, file locations in the server system, sizes of the caches and their degree of fullness. Access rates of the files also need to be recorded to enable request rate based location of files when the servers are running out of cache memory.

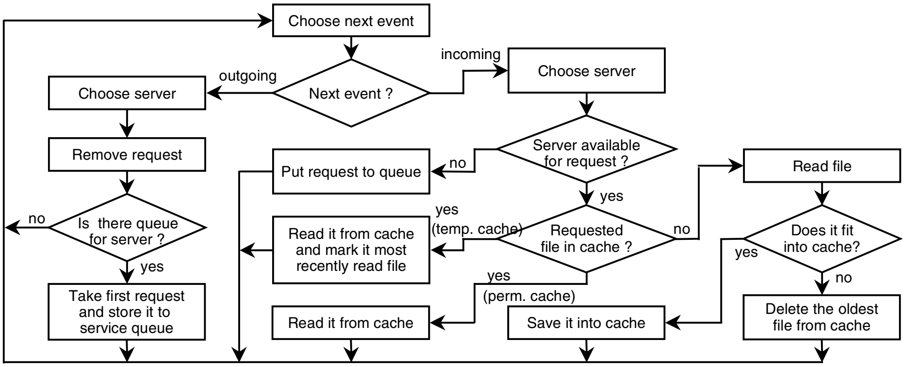


Fig. 1. Flow chart of the simulation model.

3. Simulation Examples

To demonstrated performance of the developed caching system, simulation results for a system of three servers are introduced. When a new file is inserted into the caching system, it is associated with one of four possible QoS classes and it is given a size that belongs to one of three possible categories: 1, 5 or 10 kilobytes (kb). Ten per cent (%) of the files belong to the highest QoS class (QoS1), 20 % to the second highest (QoS2), 30 % to the next (QoS3) and the rest 40 % to the lowest class (QoS4).

The simulated files were divided into two equally large groups based on their mean request rate; the higher rate was twice that of the lower one. Request rate intensities were exponentially distributed. In order to keep the simulation times reasonable, the simulated system had only 300 files. These were located randomly to servers giving approximately 100 files (about 550 kb) per server. The objective was to study performance of the proposed caching system by analyzing variation of the cache hit rates of the different QoS classes as a function of the cache size and thereby estimate the system response time. The cache sizes were varied between 0 and 550 kb, while the number of files and their sizes on each server were kept fixed.

As a comparison, performance of a non-QoS aware caching scheme, simulated in [9], was analyzed. The configuration set-up of the non-QoS aware system and the number of files, their sizes, request rates and locations were identical with those of the simulations of the QoS aware system. The only difference was the applied caching method, which did not account QoS. Instead, it exercised first-in-first-out (FIFO) discipline. At the start of a simulation, files on each server were randomly located to cache and on hard disk. The cache performed like a FIFO memory in which the most recently requested files were located at the end and less recently requested ones at the head of the FIFO queue. Each time a file (either on hard disk or in cache) was requested, it was moved to end of the FIFO. Files already in the cache were shifted towards the head of the FIFO queue. When the FIFO was full, the file at the head of the queue was removed to the hard disk.

Fig. 2 and 3 illustrate performance of the proposed QoS aware caching scheme when the proportion of files (of each QoS class) that could be located to the cache was 80 % (p_c = 0.8) and 100 % (p_c = 1.0). Fig. 4 shows corresponding results for the non-QoS aware system and Fig. 5 demonstrates the average response delay performance of these three simulated cases. The horizontal axis in all these figures gives the percentual size of the cache compared to the aggregate size of all files in the system. In Fig. 2 to 4, the vertical axis gives the percentual proportion of cache hits compared to the total number of simulated file requests. In Fig. 5 the vertical axis gives response delay normalized to file reading delay from hard disk. File reading delay from hard disk was assumed to be ten times that from cache.

Fig. 2 and 3 show that the cache hit rates of the different QoS classes follow the prefixed priorities quite nicely. The highest QoS classes reach the 100 % cache hit rate limit faster in Fig. 3 than in Fig. 2. The reason for this is that the system in Fig. 2 can store only 80 % of files of the highest QoS classes to the cache when the cache size is small. The rest of the highest QoS class files can be located to the cache only if the cache is large enough to include more than 80 % of files of all the QoS classes.

When comparing curves in Fig. 2 and 3 with those in Fig. 4, it is obvious that the proposed caching method is capable of supporting QoS. The non-QoS aware system does not show much difference between the QoS classes. The reason for the slightly differing curves is that the simulation tool assigned a QoS class, file size and request rate group randomly to every file. Thus the total size of files in each QoS class and the sizes of the two request rate groups were not exactly the above given ones.

The average response delay (see Fig 5) of the non-QoS aware system was always better than that of the QoS aware system. The reason for this is that in the QoS aware system files of the highest QoS classes have (regardless of their request intensity) priority over the lower QoS class files in locating files to cache. This lowers the average cache hit rate and lengthens the average response delay.

Simulations have shown that the performance gap between the QoS and non-QoS aware system can be reduced by decreasing the difference between the lowest and highest request rate value or decreasing the value of p_c. Allowing p_c to change step by step with the size of cache, it is possible to share cache memory fairly between the QoS classes, offer relatively good system level response delay and still maintain QoS awareness.

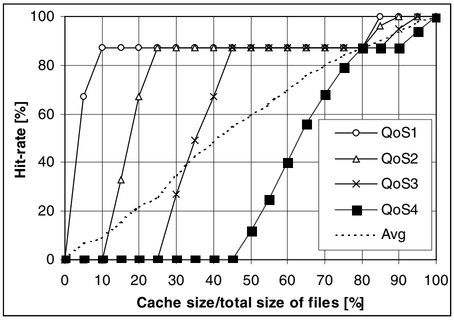


Fig. 2. Cache hit-rates of QoS aware system ($p_c = 0.8$)

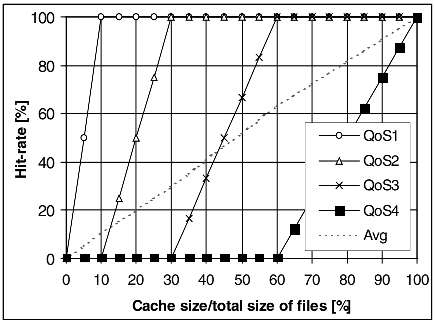


Fig. 3. Cache hit-rates of QoS aware system ($p_c = 1.0$)

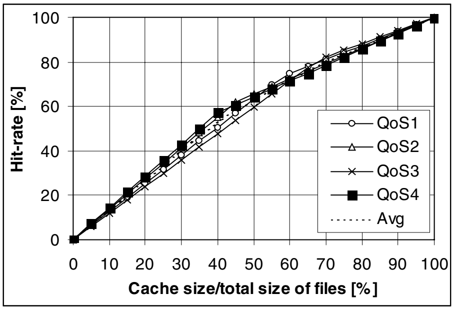


Fig. 4. Hit-rates of non-QoS aware system

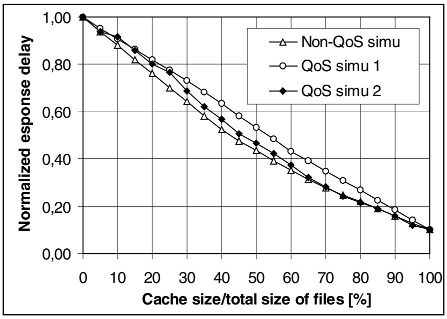


Fig. 5. Response delays of simulated systems

4. Conclusions

This paper presents a quality of service based caching scheme that allows support of different QoS classes offering shorter response delay for higher QoS class requests than for lower class ones. Processing power of the server system is utilized effectively by locating web-files randomly on the different servers and thus dividing the processing load uniformly among the servers.

The developed caching scheme was modeled by a generic simulation tool, which had previously been developed by the author, and was here used to evaluate performance of the QoS aware caching system. The model includes a number of adjustable parameters that can be varied to find optimal performance in different simulation cases. As a comparison a non-QoS aware system was also modeled to find out possible pros and cons of the developed QoS aware scheme.

The carried out simulations showed that the QoS aware caching method is clearly able to support different QoS classes. The only drawback was found in the system level cache hit rate. Since the highest QoS class files were located first to the cache memories, the cache included also less frequently requested files and the average cache hit rate was found to be lower than in the comparative non-QoS aware system. However, the discovered difference was an acceptable one. The average response delay can be lower by decreasing the portion of the highest QoS class files that can be located to the cache memory thus giving room for the lower QoS class files.

It is for further study, to enhance the developed caching scheme to implement a more efficient content distribution algorithm. The objective is to add some feedback to the algorithm and let the allocation parameters to be adjusted dynamically to respond better to changes in the servers' conditions.

References

[1] Blaze M., Alfonso R.: Dynamic Hierarchical Caching in Large-Scale Distributed File Systems. In: Proceedings of International Conference on Distributed Computing Systems, Yokohama (Japan), June 1992, pp. 521-528.

[2] Dahlin M.D., Wang R., Anderson T. E., Patterson D.: Cooperative caching: Using remote client memory to improve file system performance. In: Proceedings of Operating Systems Design and Implementation Symposium, Monterey (USA), Nov. 1994, pp. 267-280.

[3] Dan A., Towsley D.: An approximate analysis of the LRU and FIFO buffer replacement schemes. In: ACM SIGMETRICS, May 1990, pp. 143-152.

[4] Feeley M., Morgan W., Pighin F., Karlin A., Levy H., Thekkath C.: Implementing global memory management in a workstation cluster. In: Proceedings of the 15th ACM Symposium on Operating Systems Principles, Colorado (USA), Dec. 1995, pp. 201-212.

[5] Patterson R. H., Gibson G. A., Ginting E., Stodolsky D., D. Zelenka D.: Informed Prefetching and Caching. In :Proceedings of the 15th ACM Symposium on Operating System Principles, Colorado (USA), Dec. 1995, pp. 79-95.

[6] Chou H., DeWitt D.: An evaluation of buffer management strategies for relational database systems. In: Proceedings of the 11th VLDB Conference, Stockholm (Sweden), August 1985, pp. 127-141.

[7] O'Neil E. J., O'Neil P. E., Weikum G.: The LRU-k page replacement algorithm for database disk buffering. In: Proceedings of International Conference on Management of Data, Washington D.C. (USA), May 1993, pp. 297-306.

[8] Cao P., Felten E. W., Li K.: Application-controlled file caching policies. In: Proceedings of 1994 Usenix Summer Technical Conference, June 1994, pp. 171-182.

[9] Hämäläinen T., Wikström M., Raatikainen P.: A Simulation Model for Studying of Caching Algorithms. In: Proceedings of International Conferences on Info-tech and Infonet (ICII 2001), Beijing (China), Nov. 2001, pp. 599 - 604..

[10] Apostopoulos G., Aubespin D., Peris V., Pradhan P., Saha D.: Design, Implementation and Performance of a Content-Based Switch. In: Proceedings of INFOCOM 2000, IEEE, pp 1117 – 1126.

Call Admission Control for Multimedia Cellular Networks Using Neuro-dynamic Programming

Sidi-Mohammed Senouci[1], André-Luc Beylot[2], and Guy Pujolle[1]

[1] Laboratoire LIP6
Université de Paris VI
8, rue du Capitaine Scott
75015 Paris – France
{Sidi-Mohammed.Senouci, Guy.Pujolle}@lip6.fr
[2] ENSEEIHT - IRIT/TeSA Lab
2, rue C. Camichel - BP7122
F-31071 Toulouse Cedex 7 - France
andre-luc.beylot@enseeiht.fr

Abstract. We consider, in this paper, the call admission control (CAC) problem in a multimedia cellular network that handles several classes of traffic with different resource requirements. The problem is formulated as a Semi-Markov Decision Process (SMDP) problem. It is too complex to allow for an exact solution for this problem, so, we use a real-time neuro-dynamic programming (NDP) [Reinforcement Learning (RL)] algorithm to construct a dynamic call admission control policy. A broad set of experiments shows the robustness of our policies compared to the classical solutions such as Guard Channel

1 Introduction

The increasing demand and rapid growth of mobile communications that will provide reliable voice and data communications has massively grown. The service area in these networks is partitioned into cells. Each cell is assigned a set of channels[1]. As a user moves from one cell to another (handoff), any active call needs to be allocated a channel in the destination cell. If the destination cell has no available channel, the call is aborted. One of the goals of the network designer is to keep the handoff blocking probability low. If this task is simple in a mono-class traffic framework, it is quite complicated in a multi-class context. In a multi-class context it is sometimes preferable to block a call of a less valuable class and to accept another call of a more valuable class.

This paper proposes an alternative approach to solve the call admission control (CAC) in multimedia cellular networks using the experience and knowledge that could be gained during real-time operation of the system. The optimal CAC policy is obtained through a form of reinforcement learning algorithm known as Q-learning [1]. This policy is able to reduce the blocking probability for handoff calls and, also, able to generate higher revenues.

[1] Channels could be frequencies, time slots or codes depending on the radio access technique

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 1208-1213, 2002.

The rest of the paper is organized as follows. After the formulation of the CAC problem as an SMDP in section 2, we detail the two different implementations of Q-Learning algorithm (TQ-CAC and NQ-CAC) that solves this SMDP in section 3. Performance evaluation and numerical results are exposed in section 4. Finally, section 5 summarizes the main contributions of this work.

2 Problem Description

We propose an alternative approach to solving the call admission control problem in a cellular network. The approach is based on the judgment that the CAC can be regarded as a Semi-Markov Decision Process (SMDP), and learning is one of the effective ways to find a solution to this problem [3], [4], [5], [6]. In dynamic programming, we assume that the learner agent exists in an environment described by a set of possible states $S=\{s_1,s_2,...,s_n\}$. It can perform any of possible actions $A=\{a_1,a_2,...,a_m\}$ and receives a real-valued reward $r_i=r(s_i,a_i)$ indicating the immediate value of this state-action transition.

For the CAC problem, we identify the system states *s*, the actions *a* and the associated rewards *r* as follows:

1. **States:** We consider two classes of traffic *C*1 and *C*2. But, the ideas in this paper can be extended easily to several classes of traffic as well. We define the state of the system *s=(x,e)* as:
   - $x=(x_1, x_2)$ where x_1 and x_2 are the number of calls of each class of traffic (*C1* and *C2* respectively) in the cell. We do not take into account the states associated with a call departure because no action needs to be taken.
   - $e\in$ {1 =*arrival of a new C1 call*, 2 =*arrival of new C2 call*, 3 =*arrival of a C1 handoff call*, 4 =*arrival of a C1 handoff call*}
2. **Actions:** Applying an action is to accept or reject the call a∈ {1=*accept*, 0=*reject*}
3. **Rewards:** The reward *r(s,a)* assesses the immediate payoff incurred due to the acceptation of a call in state *s*. We set the reward parameters, as shown in Table 1, for each class of traffic. To prioritize handoff calls, larger reward values have been chosen for handoff calls. $r(s,a)=\begin{cases}\eta_i & if\ a=1\ and\ e=e_i\\ 0 & otherwise\end{cases}$

Table 1. Immediate Rewards

η_1	η_2	η_3	η_4
5	1	50	10

This system constitutes an SMDP with a finite state space S = {(x, e)} and a finite action space *A*={0,1}. To solve this SMDP, a particular learning paradigm has been adopted known as *reinforcement learning (RL)*. There exists a variety of RL algorithms. A particular algorithm that appears to be suitable for the CAC task is called Q-learning [1].

The task of the agent is to learn a policy, $\pi : S \rightarrow A$, for selecting its next action $a_t=\pi(s_t)$ based on the current state s_t, that maximizes the long-term revenue/utility.

For a policy π, the state-action value $Q^{\pi}(s,a)$ (named Q-value) is the expected discounted reward for executing action a at state s and then following policy π thereafter. The Q-learning process tries to find the optimal Q-values in a recursive manner. The Q-learning rule is

$$Q_{t+1}(s,a) = \begin{cases} Q_t(s,a) + \alpha_t \Delta Q_t(s,a), & if \;\; s = s_t and \;\; a = a_t \\ Q_t(s,a), & otherwise \end{cases} . \quad (1)$$

$$\text{Where } \Delta Q_t(s,a) = \left\{ r_t + \gamma \max_b \left[Q_t(s'_t, b) \right] \right\} - Q_t(s,a) . \quad (2)$$

3 Algorithm Implementation

After the specification of the states, actions and rewards, let us describe the two online implementations of the Q-learning algorithm for solving the CAC problem (TQ-CAC and NQ-CAC). The TQ-CAC uses a lookup table to represent the Q-values. In contrast, the NQ-CAC uses a multi-layer neural network. Function approximators such as neural networks are used when the input space consisting of state-action pairs is large or the input variables are continuous.

When there is a call arrival (new or handoff call), the algorithms determine the action according to

$$a = \arg \max_{a \in A(s) = \{0,1\}} Q^*(s,a) \, . \quad (3)$$

In particular, (3) implies the following procedures. When a call arrives, the Q-value of accepting the call and the Q-value of rejecting the call are determined. If rejection has the higher value, the call is dropped. Otherwise, the call is accepted.

In these two cases, and to learn the optimal Q-values $Q^*(s,a)$, the value function is updated at each transition from state s to s' under action a for the two algorithms as follows:

1. TQ-CAC: (1) is used to update the appropriate Q-value in the lookup table.
2. NQ-CAC: In this case, ΔQ defined in (2) is served as an error signal which is backpropagated in the *back-propagation (BP)* algorithm [1].

We compare our policies with the greedy policy[2] and with the Guard Channel mechanism [2]. The number of guard channels is determined for each traffic period and each traffic class. The guard channel mechanism will be characterized by a vector s which corresponds to the different thresholds, $s = (s_1, s_2, ..., s_K)$, where K is the number of classes of traffic. In the present paper, an exact numerical solution has been derived. To determine the optimal vector s^*, all the configurations s for which $s_1 \le s_2 \le ... \le s_K = N$, where N is the number of channels in the cell, were investigated.

[2] Policy that always accepts a call if the capacity constraint will not be violated

4 Simulation

In order to evaluate the benefits of our call admission control algorithms, we simulate a mobile communication system using a discrete event simulation. We consider a fixed channel assignment (FCA) system with *N=24* channels in each cell. The performance of the algorithms has been evaluated on the basis of the total rewards of the accepted calls (*Total rewards*), the total rewards of the rejected calls (*Total Lost Rewards*), and by measuring the handoff blocking probability.

A set of simulations was carried out, including the cases of traffic load varying, and time-varying traffic load. The experimental results are shown in Fig. 1 through Fig. 3. The results show that the reinforcement learning is a good solution for the call admission control problem. The proposed algorithms are considerably powerful compared to the greedy and to the guard channel schemes. In all cases the lost rewards due to rejection of customers and blocking probability of handoff calls are significantly reduced. The total rewards due to acceptance of customers is also significantly increased.

The Q-values were first learned during a training period with a constant traffic load for both *C*1 and *C*2. The parameters used in the simulation are given in Table 1 and 2.

Table 2. Experimental Parameters

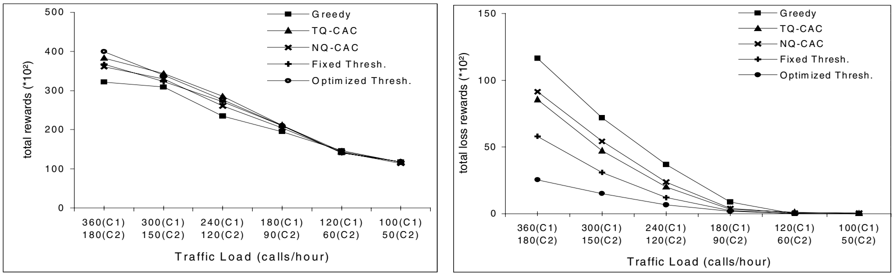
	C1	C2
Number of channels	1	2
Call holding time	40 s	40 s
Call arrival rate	$\lambda_1 = 180$ *calls / hour*	$\lambda_2 = \lambda_1 / 2 = 90$ *calls / hour*

4.1 Traffic Load Varying

In this case we used the same policy learned in the training period but with six different traffic load conditions (for both classes C1 and C2). Fig. 1 and Fig. 2 show that the proposed algorithms result in significant gains compared with alternative heuristics for all the considered traffic loads and especially when the traffic load is heavy.

It is shown that TQ-CAC leads to significantly better results compared to NQ-CAC because NQ-CAC uses a neural network to represent the Q-values which needs more time to converge.

We also compare our algorithms results to those obtained with: (1) the guard channel with fixed thresholds – these thresholds were calculated for the same traffic load given in Table 2; (2) the guard channel with optimized thresholds - the best thresholds are derived for each input traffic load value.



(a) (b)

Fig. 1. (a) Total rewards per hour (b) Total Loss rewards per hour

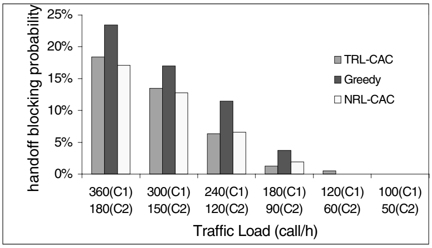


Fig. 2. Handoff blocking with six different traffic loads

This illustrates clearly that TQ-CAC and NQ-CAC have the potential to significantly improve the performance of the system over a broad range of network loads.

We notice in Fig. 1, that the optimal threshold method leads to better performance results than Q-learning. But in this method we must compute the optimal values for each traffic in an off-line manner. In contrast, in TQ-CAC and NQ-CAC, it is interesting to observe that neither the table nor the neural network were relearned and retrained for each traffic load, indicating that the system possesses some generalization and adaptability capabilities.

4.2 Time-Varying Traffic Load

The traffic load in a cellular system is typically time varying. In this case, we use the same policy learned in the training period but during a typical 24-h business day. The peak hours occur at 11:00 a.m. and 4:00 p.m. Fig. 3 gives the simulation results under the assumption that the two traffic classes followed the same time-varying pattern. The blocking probabilities were calculated on an hour-by-hour basis. The improvements of the proposed reinforcement learning algorithms over the greedy policy are apparent specially when the traffic is heavy.

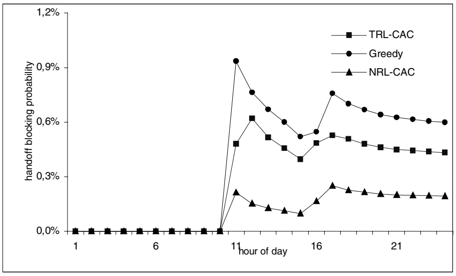


Fig. 3. Performance with time-varying traffic load

5 Conclusion

In this paper, we presented a new approach to solve the problem of call admission control in a cellular multimedia network. We formulate the problem as a dynamic programming problem (SMDP), but with a very large state space. The optimal solutions are obtained by using a self-learning scheme based on Q-Learning algorithm. The benefits gained by this method can be summarized as follows. First, the learning approach provides a simple way to obtain an optimal solution for which an exact solution can be very difficult to find using traditional methods. Second, compared to other schemes like the guard channel, the system offers a generalization capacity. So, any unforeseen event due to significant variations in the environment conditions can be considered as a new experience for improving its adaptation. Third, the acceptation policy can be determined with very little computational effort. It is, also, shown that the proposed CAC algorithms result in significant savings.

References

1. T. M. Mitchell, "Machine Learning", *McGraw-Hill companies, Inc.*, 1997.
2. C.H. Yoon, C.K. Un, Performance of personal portable radio telephone systems with and without guard channels, *IEEE Journal on Selected Areas in Communications (JSAC'1993)*, vol. 11, pp. 911-917, August 1993.
3. P. Marbach, O. Mihatsch and J. N. Tsitsikils, "Call admission control and routing in integrated services networks using neuro-dynamic programming", *IEEE Journal on Selected Areas in Communications (JSAC'2000)*, vol. 18, N°. 2, pp. 197 –208, Feb. 2000.
4. H. Tong and T. X. Brown, "Adaptive Call Admission Control under Quality of Service Constraint: a Reinforcement Learning Solution", *IEEE Journal on Selected Areas in Communications (JSAC'2000)*, vol. 18**,** N°. 2, pp. 209-221, Feb. 2000.
5. R. Ramjee, R. Nagarajan and D. Towsley, "On Optimal Call Admission Control in Cellular Networks", *IEEE INFOCOM*, pp. 43-50, San Francisco, CA, Mar. 1996.
6. S. Senouci, A.-L. Beylot, Guy Pujolle, "A dynamic Q-learning-based call admission control for multimedia cellular networks", IEEE International Conference on Mobile and Wireless Communications Networks (MWCN'2001), pp. 37-43, Recife, Brazil, Aug. 2001.

Aspects of AMnet Signaling

Anke Speer, Marcus Schöller, Thomas Fuhrmann, and Martina Zitterbart

Universität Karlsruhe, Germany

Abstract. AMnet provides a framework for flexible and rapid service creation. It is based on Programmable Networking technologies and uses active nodes (AMnodes) within the network for the provision of individual, application-specific services. To this end, these AMnodes execute service modules that are loadable on-demand and enhance the functionality of intermediate systems without the need of long global standardization processes.

Placing application-dedicated functionality within the network requires a flexible signaling protocol to discover and announce as well as to establish and maintain the corresponding services. AMnet Signaling was developed for this purpose and will be presented in detail within this paper.

Keywords: Programmable Networks, Active Nodes, Multicasting, Signaling

1 Introduction

Many new evolving Internet applications are based on one-to-many or many-to-many communication, e.g., tele-teaching, tele-collaboration, information dissemination through push technologies, and web-radio. IP multicast [2] efficiently supports this type of transmission with a receiver-oriented concept: receivers join a particular multicast session group and traffic is delivered to all members of that group by the network infrastructure.

A challenge that comes with this type of communication is the possible *heterogeneity* in the group members' service requirements. These may vary dependent on the individually available performance of the network access, the type of end system being used, the willingness to pay a higher price for better quality of service and the like. Today, most approaches realizing heterogeneous group communication adjust the provided data stream for all group members according to the group member with the lowest performance. This, however, is not desirable for many multimedia or distributed applications (e.g., video conferencing, gaming).

Besides group communication applications, also other Internet applications benefit from additional network support (e.g., management or control facilities). However, introducing new functionality into the network has to be in-line with new evolving applications for realizing proper communication support promptly. Unfortunately, progress in supporting new network functionality is very slow

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 1214–1220, 2002.

because the current network infrastructure is inflexible. The introduction of new services and protocols to enhance network functionality typically requires long global standardization processes.

AMnet addresses *rapid service creation* in the context of heterogeneous group communication to allow the flexible and rapid introduction of new functionality in global networks. AMnet is based on Programmable Networking technologies and aims at building an implicit overlay network on top of the existing IP infrastructure for the completion of application-specific requirements. According to the Programmable Networking approach, so-called *service modules* are installed on active intermediate nodes – the *AMnodes*. AMnodes form the core building blocks of AMnet and operate on the multicast distribution tree used for the communication between sender and receivers. Service modules are responsible for the adaptation of data streams to specific service demands [5].

This paper is structured as follows: The next section presents the developed inter-domain signaling protocol for AMnet in detail. Section 2.1 describes the management of different services within AMnet, followed by the mechanisms of the establishment and maintenance of these services in Section 2.2. In Section 2.3, the way a receiver is provided with its dedicated services is presented. Section 2.4 lays special focus on the new concepts developed for evaluating AMnodes to determine their capabilities as possible service providers. The paper closes with a summary and an outlook on future work.

2 Concepts of AMnet Signaling

AMnet aims at dynamically placing application-specific functionality within the network. To this end, some questions have to be decided: how should different services be managed within a *session*, how should they be established and maintained, how should a receiver be associated to a dedicated service and where should those services be placed? In this context, a session describes a communication scenario where a designated sender issues a data stream which can be received from several communication participants without or after adaptation in the AMnodes. To solve the foregoing questions a flexible and light-weight signaling protocol for AMnet was developed [6] which will be presented in the following section.

2.1 Management of Services

Service heterogeneity within a session needs to be bound to a manageable degree of diversity. Therefore, one concept of AMnet signaling is to logically group receivers with similar service demands into distinct multicast receiver groups – the *service level groups* – distinguishable by their multicast-addresses. The receivers join the corresponding group on demand through IGMP [3].

Each service level group within a communication session represents all receivers whose service demands can be satisfied with a single group service.

Therefore, each group represents a different view onto the same original data corresponding to the adaptation performed by the AMnodes.

The service of a group is supported by an AMnode through the use of appropriate service modules. The actual service is then derived from the processing of the original data stream (cf., AMnode 2 in Figure 1) or from the service of another service level group (cf., AMnode 3 in Figure 1). Therefore, the communication service offered by AMnet can be described by a tree of service level groups (cf., Figure 1).

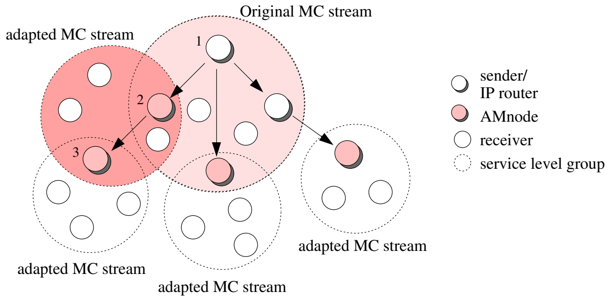


Fig. 1. Multicast Tree with Service Level Groups

2.2 Establishment and Maintenance of Services

Service modules are held in distributed data bases – the *service module repositories* – which are administratively managed per domain. Service modules can be stored there by trusted AMnet users or network management procedures. Moreover, current work focuses on establishing a hierarchy of trusted repositories. The stored service modules are grouped into module classes like, e.g., audio transcoding or reliable multicast, and for each module class there exists a distinct evaluation procedure to be downloaded within the evaluation process (cf., Section 2.4). Therefore, the overall purpose of the repositories is to make service modules and their corresponding evaluation procedures available to an AMnode which is requested to provide a special service (cf., Figure 2 (4)).

Besides multicasting the original data stream of the session (cf., Figure 2 (1)), the sender announces the provided session on a separate multicast group – the *session control group* (cf., Figure 2 (1a)). In this group every AMnet session is announced similar to the Session Announcement Protocol of the MBone [4]. The session announcement contains a description of the session including bandwidth and delay requirements, as well as content specific information like data format and compression scheme. This description is used by the potential session participants to determine in which way the original data stream has to be adapted by the AMnodes to receive the data stream at a desired service level.

Moreover, the description contains the multicast address of the original data stream and the multicast address of the *service announcement group*. In this group, the AMnodes announce the provided services (cf., Figure 2 (5)). Potential session participants join the service announcement group to learn about available services (cf., Figure 2 (3)), i.e., the description of the provided service, as well as the address of the corresponding service level group where the adapted data is sent to. Moreover, during multicast distribution the address of each AMnode that was traversed by the announcement on its way from the sender to the receiver is included. This information is used for the evaluation process described in Section 2.4.

Service modules on an AMnode are maintained in a soft state. After joining a service level group the participant periodically sends HELLO messages to the AMnode hosting the appropriate service module. If no HELLO messages were received for a given time, the AMnode makes the service module stop issuing data into the corresponding service level group. However, the service modules are not immediately deleted from the AMnodes but cached, in case the service is requested again right afterwards. The soft state of the service modules is utilized to prevent service level groups not used by any participants. The caching strategy helps to avoid unnecessary overhead coming along with re-loading and re-installation.

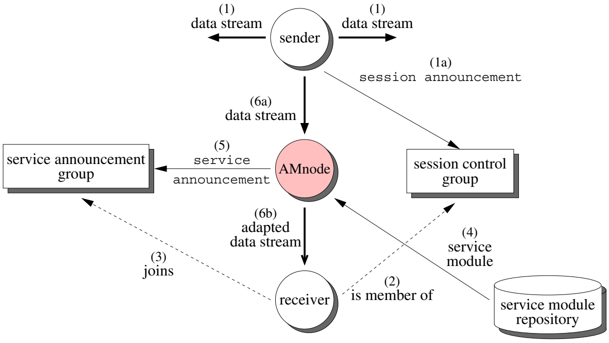


Fig. 2. Overview of the AMnet Service Control

2.3 Association between Receiver and Service

A receiver that wants to use a special service for adapting the data stream to its requirements processes the service announcements. If one of the announcements matches the receiver's requirements, it simply joins the corresponding service level group where the adapted data stream is sent to.

Otherwise, if no matching service is announced, an *evaluation process* has to be started (cf., Section 2.4). Within this process, AMnodes on the data path between the sender and the receiver are analyzed whether to be capable of providing the desired service. The AMnodes on the data path are known from the corresponding service announcements of the sender that provides the data stream the receiver wants to be adapted (cf., Section 2.2). Even if no additional service is advertised in the service announcement group, at least the service of the session sender providing the original data stream is announced.

The evaluation process will result in the address of an AMnode that is considered to be a good place for supporting the service. Then, the appropriate service module is downloaded from the service module repository onto this AMnode (cf., Figure 2 (4)). For security reasons, the service modules will be signed and only modules with a correct signature will be installed on an AMnode. The newly installed service is announced into the service announcement group (cf., Figure 2 (5)) and the receiver can simply join the corresponding service level group. Now, it will experience the data stream adapted according to its requirements (cf., Figure 2 (6a)-(6b)).

2.4 The Evaluation Process

In the original approach of the evaluation process for AMnet [6] the intra-domain evaluation of the AMnodes was realized with active evaluation packets corresponding to capsules as introduced in the Active Networking context [1]. These capsules contained an evaluation program downloaded from the service module repository and initialized by the receiver. This approach, however, was not usable within the context of inter-domain signaling because of security considerations. Therefore, a new approach was developed. Now, the receiver has only to issue a *service request* to its predecessor AMnode (cf., Figure 3) known by the path information of the service announcements. The service request contains the class of the service the receiver wants to be supported (e.g., audio transcoding) and the path information. Moreover, specific parameters can be included. In the case of audio transcoding, this may be the maximum data rate the receiver is able to process, the formats of the data stream the receiver can understand, and so on.

The first AMnode that is able to process the receiver's service request contacts the service module repository and downloads the evaluation procedure that corresponds to the module class the desired service belongs to. With this evaluation procedure the AMnode is analyzed. The results of the local evaluation, the address of this local AMnode, as well as the identifier for the used evaluation procedure are tied up into an evaluation packet that is forwarded to the next AMnode (cf., Figure 3) known from the path information contained in the service request. Moreover, this path information is included into the packet, as well as the address of the first AMnode that processed the receiver's service request and started the whole evaluation process.

An AMnode that receives an evaluation packet contacts its responsible service module repository to download the specified evaluation procedure. The AMnode will contact the same service module repository for downloading a requested service. Therefore, if the specified procedure is not included in the contacted service module repository, the AMnode will not be able to support the desired service at all, because service class and corresponding evaluation procedure are always stored as one entity in a repository entry. Then, the evaluation packet will only be forwarded unchanged to the next AMnode on the path towards the session sender. However, if the contacted service module repository contains the specified evaluation procedure, it will be downloaded and executed on the AMnode as described above. Evaluation results are only entered into the evaluation packet if the local AMnode fulfills the given requirements in a better way than the AMnode already registered in the packet. After the evaluation procedure is finished on the local AMnode, the evaluation packet is forwarded to the next AMnode, again. This process stops on the last AMnode in front of the session sender. After its evaluation, the final evaluation packet is sent back to the first AMnode that started the whole procedure (cf., Figure 3). This AMnode, now, interprets the evaluation result as the address of the AMnode that is considered to be the best for supporting the service in the current network scenario. This AMnode, then, is made to download and install the requested service module and the receiver can access the desired service level as described in Section 2.3.

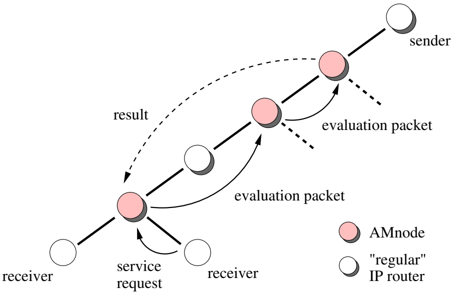


Fig. 3. Scheme of the Evaluation Process

3 Conclusions and Outlook

AMnet provides an open and generic framework for the provision of user-tailored rapid service creation with a specific focus on heterogeneous group services. It is based on Programmable Networking technologies and aims at building an overlay network on top of the available Internet infrastructure. A major goal of AMnet is to provide individual services on demand without complex installation and management overhead. AMnet is based on IP and benefits from its multicast extensions in several ways. For realizing service and session announcements, as well as for disseminating adapted data with individual requirements distinct IP multicast groups are applied.

This paper is focused on AMnet Signaling – a flexible and active signaling protocol – developed specifically to support the placement and announcement as well as the establishment and maintenance of active services in the context of rapid service creation with AMnet. Moreover, the new developed evaluation process for placing dedicated services inside the network was described. In contrast to the mechanisms presented in [6], this evaluation mechanism can be used inter-domain.

Future work will focus on extending or, respectively, changing the signaling mechanisms to be able to use AMnet as well in networking environments where native IP Multicast is not provided. Different approaches are considered and will be evaluated. Moreover, the presented novel evaluation process will be introduced in the actual prototype implementation, leading to enhanced experience with automated service discovery and placement.

References

1. J. V. Guttag D. J. Wetherall and D. L. Tennenhouse. ANTS: A Toolkit for Building and Dynamically Deploying Network Protocols. In *Proceedings of the IEEE OPENARCH*, pages 117–129, April 1998.
2. S. Deering. Host Extensions for IP Multicasting. RFC 1112, IETF, August 1994.
3. B. Fenner. Internet Group Management Protocol, Version 2. RFC 2236, IETF, November 1997.
4. M. Handley. SAP: Session Announcement Protocol. Internet draft, IETF, November 1996.
5. T. Harbaum, B. Metzler, R. Wittmann, and M. Zitterbart. AMnet: Heterogeneous Multicast Services based on Active Networking. In *Proceedings of the IEEE OPENARCH99*, pages 98–107, New York, NY, USA, March 1999. IEEE.
6. A. Speer, R. Wittmann, and M. Zitterbart. Locating Services in Programmable Networks with AMnet Signalling. In *Proceedings of The Sixth Conference on Intelligence in Networks (SmartNet 2000)*, Wien, Austria, September 2000.

Virtual Home Environment for Multimedia Services in 3rd Generation Networks

Orazio Tomarchio, Andrea Calvagna, and Giuseppe Di Modica

Dipartimento di Ingegneria Informatica e delle Telecomunicazioni
Università di Catania
Viale A. Doria 6, 95125 Catania, Italy
{tomarchio, acalva, gdmodica@diit.unict.it}

Abstract. *The Virtual Home Environment (VHE) has been introduced as an abstract concept enabling users to access and personalize their subscribed services whatever the terminal they use and whatever the underlying network used. The European IST VESPER project (Virtual Home Environment for Service Personalization and Roaming Users) aims to provide an architectural solution and an implementation of the VHE, providing ubiquitous service availability, personalised user interfaces and session mobility, while users are roaming or changing their equipment. In this paper we present a multimedia delivery service, one of the trial services selected to demonstrate VHE features, showing its interconnection with the so far defined VESPER VHE architecture.*

1 Introduction

The technological evolution of the last years, both in network speed and bandwidth than in multimedia capabilities of low-end devices, has made possible the convergence of telecommunication networks and data networks, leading to a new generation of integrated, IP based, transport infrastructure that will enable the deployment of even more valuable services for the users, like real-time video communication ones. Also, both existing and upcoming wireless technologies are enabling the support of data services and audio/video communication for "moving" users, that is users whom network location may change even while a service session is currently in progress. As these services will be available over heterogeneous network, users would like to access them in a personalized way, transparently and independently of the underlying network technology and particular terminal used.

In 3GPP [1] this idea is embodied in the Virtual Home Environment (VHE)[2,3,4], defined as a concept for Personal Service Environment (PSE) portability across network boundaries and between terminals. The concept of the VHE is such that users are consistently presented with the same personalized features of subscribed services, in whatever network and whatever terminal (within the capabilities of the terminal and the network), wherever the user may be located. The IST VESPER (*Virtual Home Environment for Service Personalization and Roaming Users*) project [12] (funded by the European Community) aims to provide an architectural solution and an implementation of the VHE, providing ubiquitous service availability, personalized user interfaces (i.e., service portability) and session mobility, while users are roaming

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 1221-1226, 2002.

or changing their equipment. VESPER VHE should hide away from the user the variety of access network types (fixed or wireless), the variety of supporting terminals, and the variety of the involved network and service providers involved during service provision [9,10]. As regards the international standardization for a, the project intends to influence the course of standardisation within 3GPP, Parlay [8] (to enhance standard APIs for sustaining the VHE functionality) and OMG [5,6].

In the context of the Vesper project, this paper describes our effort in the realization of a test service-application [1226] to validate the design and implementation work done in the project [11]. The service we will describe, selected among others as one of the trial services used to demonstrate the features of the VESPER prototype, is a "multimedia content delivery" service, designed to provide a mechanism to distribute multimedia streams (consisting of video and audio, but also pictures, etc) to end-users. In this paper we will show the main benefit gained by such a kind of service (which we named "*multimedia delivery service*"(MDS)), when used in the context of a Virtual Home Environment. The rest of the paper is organized in the following way. Section 2, after a brief overview of the Vesper VHE architecture, describes the multimedia delivery service and its interactions with the VHE architecture at the current phase of the project. In Section 3 we focus on the adaptation feature of VHE and finally, we conclude the paper in Section 4.

2 Vesper Demonstration Services

The VESPER architecture has been designed using a component based approach: all of these components rely on a CORBA based environment for their internal communication. A more detailed description of the overall VESPER architecture is out of scope of this paper and can be found in [10,12]. However, VESPER components are embedded into a heterogeneous network and terminal landscape. Figure 1 shows VHE architectural placement in relation to network and terminal environment. At server side VHE functionality is accessed via VHE API on top and deals via OSA/Parlay[8] gateways with different networks as transport layer. At terminal side VHE functionality is also accessed via VHE API and deals via USAT (Universal SIM Application Toolkit)[14] or MExE (Mobile Station Application Execution Environment)[13] with terminal core functions.

One of the objective of the VESPER project is to define, design, and implement services which both impose precise requirements on the VHE architecture defined and implemented by the project, and demonstrate that this VHE architecture fulfils the requirements. VESPER will provide an open API to VASPs' applications, enabling the VHE concept within the service. VASPs (*Value Added Service Provider*) will be able to offer advanced services, abstracting from the terminal used for accessing the service and from the underlying networks, leaving all of the basic VHE services to the VHE provider role. In this scenario, this section will describe the MDS service, one of the applications selected for validating the VESPER middleware during the two trials, the first one held at the end of 2001 in a lab environment, the second one (at the end of 2002) involving also real networks.

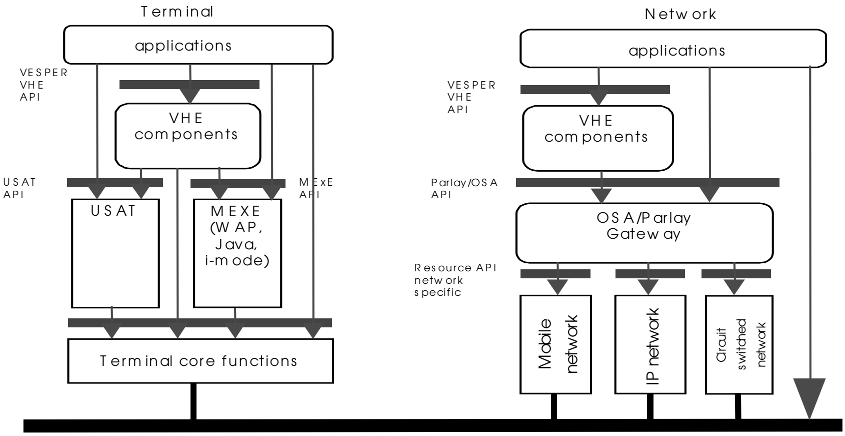


Fig. 1. Embedding of VHE Architecture

2.1 Multimedia Delivery Service

People are nowadays used to get every kind of information through the Internet, using common Web browser applications. The next big challenge is to deliver multimedia informations to end users independently of the device used to access the network: notebook, PDA, and next generation UMTS phone will be able to display not only text and images but also audio and video.

Using VHE functionalities a multimedia delivery application will be greatly enhanced and widely spread among users. Users will not be restricted in the set of terminal they have to use to access the application and to receive data, neither in the network they are connected to. Users will be able to access the application using different terminals, ranging from the powerful multimedia PC to the personal PDA with limited screen size and computing capabilities. Even future smartphones will be able to reproduce small videoclips in their small screens.

The MDS will take benefit of adaptation, connectivity and service personalisation functionalities provided by VESPER VHE. The adaptation feature will make the service delivery transparent to the VASP, thus allowing a larger number of terminals to be able to access the MDS from a wider range of available networks. The service will only provide the stream content to the VHE component responsible of the adaptation, whose task will be to adapt it to the user preferences, user network, user terminal and deliver it. What is asked to the adaptation is not only a user interface adaptation, but also a content adaptation. So, for instance, if user terminal is not provided with the right codec to watch a movie, it is a task of the component responsible for adaptation either to “adapt”(codify) the stream in a format compatible to that of one of the codecs owned by the user terminal, or to upload the suitable codec to the terminal.

3 Service Adaptation in the Vesper VHE

One of the key features of VHE, is the adaptation that it provides to terminal capabilities and user preferences. Each service using Vesper VHE APIs should not care of terminal device used by the end-user: it is the adaptation that takes charge of that. The adaptation is realized by the Adaptation Component, whose main tasks are:

- to adapt the contents a VESPER Service provides to the user according to the capabilities of the terminal accessing the VHE Server, the End-User interface and services preferences and the QoS classes supported by the underlying network(s),
- to manage the user interaction with the service.

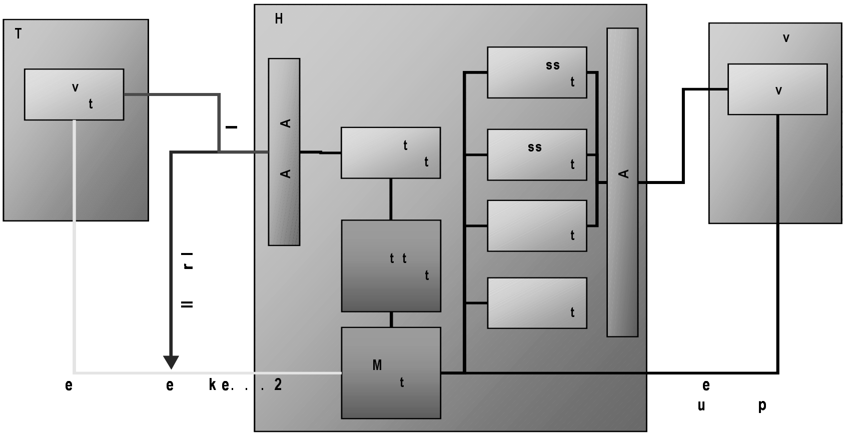


Fig. 2. Media adaptation

The VHE service implementation is completely independent from the actually used environment (network type, terminal type, user preferences) during service usage. The adaptation component offers interfaces in the VHE API to enable this feature. An overview of an adaptation scenario is given in Figure 2. The adaptation to network and terminal capabilities is done at the VHE system. The content flow goes from the VASP Server to the media adapter via TCP/IP connection through the CORBA based VHE API. On the terminal side a connection is established by the Connection Component depending on the transport network via OSA/Parlay. The media adapter supports the used network protocol to provide the terminal with the media stream (e.g. announcements, multimedia/video). The Adaptation Component specification has provisioned for a flexible engineering scheme such that the media adapter can be wrapped as a mobile agent whose itinerary is limited to the VASP Server and/or the End-User's terminal. The adaptation is done at VASP Server by the agent providing adaptation to the media stream supported by the terminal and to the used network protocol. At terminal side a corresponding agent decodes the stream and provides content output at the terminal. This solution presupposes that the terminal and the VASP Server provide an agent execution environment.

The second role of the adaptation is to manage the user interaction with the service: this means that the user interface should be adapted and presented to the user according the terminal capabilities (apart from user preferences). In order to be able to offer this kind of adaptation each service is required to provide a formal description (*UIModel*) of the user interface they want to offer to their users: this description is expressed in XML and includes several kind of logical tools for user interaction (buttons, text fields, text entries, checkbox, etc). The actual representation of this graphical model will depend of the actual device used by the user to access the service. It is the adaptation component that will decide the best way to render the User Interface model (UIModel) provided by the service.

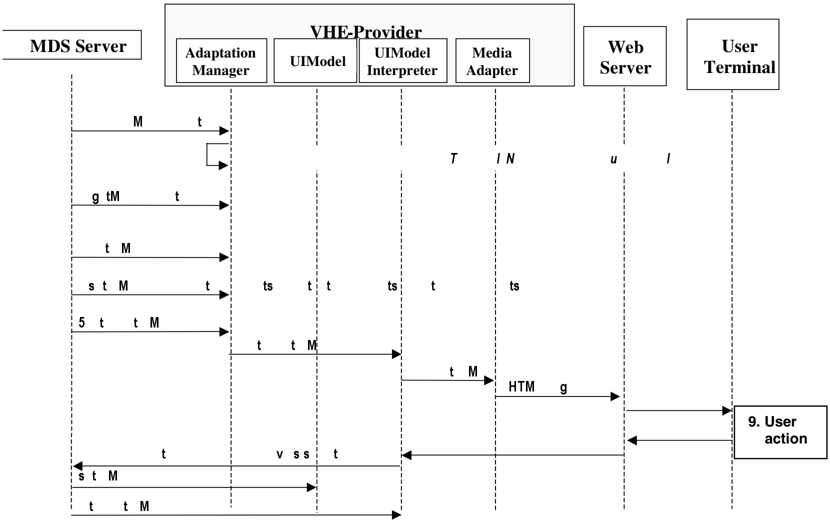


Fig. 3. MDS interaction with the VHE Adaptation component

For better understanding of this mechanism, Figure 3 shows a step by step scenario involving the MDS service interaction with the adaptation component, supposing the user has accessed the service through a Web browser:

1. The service looks up for a media adapter. The VHE Server provides the service with a list of available and appropriate media adapters for the terminal and network currently used.
2. The MDS chooses a media adapter whose presentation characteristics cope best with its logic.
3. The service asks for an UIModel object.
4. The MDS sends the description of the service's user interface to this object as an XML description. This description synthesises the interface to be presented in the End-User's terminal (output messages, input fields, selection list, buttons, etc).
5. The MDS asks the Adaptation component to interpret the UIModel.
6. (and 7) The Adaptation Component interprets the model and formats it into an HTML page, by mapping the output elements, input elements and action elements into HTML elements, respectively HTML texts, HTML text field/selection field and HTML buttons. The mapping of the previous UIModel description into HTML page takes in consideration the terminal capabilities and the End-User User Interface Preferences.

8. Once produced the adapted HTML page, the Adaptation Component submits the page to the Web server which then sends this page to user's browser.
9. The user interacts with the received HTML page, fills the text field or selects values in the selection field and then clicks on a button.
10. The Web server forwards the request to the registered UIModel Interpreter.
11. The UIModel Interpreter object collects the information in the URL request, builds a description of user's interaction (user's entered values, button pressed) in form of a XML description and invokes MDS callback action listener.

4 Conclusions

In this paper an overview of the VESPER project has been presented. The paper has been focused on the description of a multimedia delivery service, selected as a trial application for demonstrating VHE capabilities. These kind of applications can be greatly enhanced by VHE features, since users will be able to access this advanced service by means of every available device. Key functionalities of adaptation to terminal capabilities and personal user preferences have been described more in detail.

Acknowledgement. This work has been performed in the framework of the project IST VESPER, which is funded by the European Community. The Author(s) would like to acknowledge the contributions of his (their) colleagues from *Intracom Hellenic Telecommunications and Electronics Industry S.A., National Technical University of Athens, Institut National de Recherche en Informatique et Automatique, IKV++ Technologies AGe, FOKUS Fraunhofer Institute for Open Communication Systems, Fondazione Ugo Bordoni, Universita' Di Catania, Portugal Telecom Inovação, University of Surrey, Technical Research Centre of Finland* and *SIEMENS AG Österreich*

References

1. 3 rd Generation Partnership Project (3GPP), http://www.3gpp.org
2. *"Virtual Home Environment / Open Service Architecture*", TS 23.127, 3GPP project.
3. *"The Virtual Home Environment*", TS 22.121, 3GPP project, release 2000
4. 3GPP, 3G TS 22.121 v3.2.0, The Virtual Home Environment, stage 1.
5. 3GPP, 3G TS 29.198 v3.0.0, Open Service Access (OSA) API Part 1, stage 3.
6. 3GPP, 3G TS 23.127 v3.1.0 Virtual Home Environment/Open Service Architecture, st. 2.
7. A. Calvagna, A. Puliafito, and L. Vita, ``A Low Cost/High Performance Video on Demand Server'', in IEEE Int. Conf. on Computer and Communication Networks (ICCCN'99), Boston, MA USA, 11-13 Ottobre 1999
8. Parlay Group, Parlay Specifications, http://www.parlay.org/
9. VESPER, IST-1999-10825, Technical Annex
10. VESPER, IST-1999-10825, D22 – VHE Architectural Design
11. VESPER, IST-1999-10825, D42 – Initial Demonstration Services Specification
12. VESPER WWW site: http://vesper.intranet.gr/
13. 3G 22.057 v3 0 1 – MexE
14. ETSI TS 101.267 v8.3.0, Specification of the SAT for the SIM-ME interface

On Providing End-To-End QoS Introducing a Set of Network Services in Large-Scale IP Networks

E. Tsolakou, E. Nikolouzou, and S. Venieris

National Technical University of Athens
School of Electrical and Computer Engineering
Telecommunications Laboratory
9 Heroon Polytechniou Str, 15773 Athens, Greece
{evi, enik}@telecom.ntua.gr, ivenieri@cc.ece.ntua.gr

Abstract. The Differentiated Services (DiffServ) architecture has been proposed as a scalable solution for providing service differentiation among flows. Towards the enhancement of this architecture, new mechanisms for admission control and a new set of network services are proposed in this paper. Each network service is appropriate for a specific type of traffic and is realized through its own network mechanisms, which are the Traffic Classes. Traffic Classes provide the traffic handling mechanisms for each Network Service and are composed of a set of admission control rules, a set of traffic conditioning rules and a per-hop behavior (PHB). Different traffic-handling mechanisms are proposed for each network service and are implemented with the use of the OPNET simulation tool. A large-scale network is used as a reference topology for studying the performance and effectiveness of the proposed services.

Keywords: *Network Services, QoS, Traffic Classes*

1 Introduction

Motivated by the rapid change of QoS requirements of the new introduced network applications, the Internet has been evolving towards providing a wide variety of services, in order to meet the qualities of information delivery demanded by the applications. For the past few years, there have been two major efforts focusing on augmenting the single-class, best effort Internet to include different levels of guarantee in quality of service - Integrated service (Intserv) and Differentiated service (DiffServ) [1]. The most salient point between these two approaches is the difference on the treatment of packet streams. Intserv tends to emulate circuit-switch networks, focusing on guaranteeing QoS on individual packet flows between communication end-points. To ensure the level of guarantee on a per-flow basis, it requires explicit signaling to reserve corresponding resources along the path between these end-points. One major dilemma faced by this approach is that in the core of the Internet, where exist several millions of flows, it may not be feasible to maintain and control the forwarding states efficiently. These scalability and management problems are addressed recently by DiffServ approach.

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 1227-1232, 2002.

The focal point of the DiffServ model lies in the differentiation of flows at an edge router of a DS-domain and the aggregation of those flows of the same service class at a core router of the DS-domain. At each ingress interface of a edge router, packets are classified and marked into different classes, using Differentiated Services CodePoint (DSCP). Complex traffic conditioning mechanisms such as classification, marking, shaping, and policing are pushed to network edge routers. Therefore, the functionalities of the core routers are relatively simple - they classify packets and then forward them using corresponding Per-Hop Behaviors (PHBs). In this sense, PHB is a means by which a node allocates resources to behavior aggregates, and it is on top of this basic hop-by-hop resource allocation mechanism that useful differentiated services may be constructed. PHBs are implemented in nodes by means of some buffer management and packet scheduling mechanisms and the parameters associated with those mechanisms are closely related to those of traffic conditioning.

2 Network Services

In order to provide QoS guarantees in a DiffServ network it is essential to assure QoS differentiation. Therefore, a set of five Network Services (NS) has been specified and implemented in our framework [2], which comprises the services sold by the provider to the potential customers, either end-users or other providers. The specified NSs are: Premium Constant Bit Rate, Premium Variable Bit Rate, Premium Multimedia, Premium Mission Critical and Standard Best Effort.

The PCBR network service is intended to support applications that require VLL-like services, i.e. voice flows, voice trunks, interactive multimedia applications with low bandwidth requirements. That kind of flows is usually characterized by an almost constant bit rate (CBR) and low bandwidth requirements, while a great number of them are unresponsive (UDP). In addition, they should have small packets (<256Bytes), so as not to provoke long transmission delays. It requires and expects to receive low delay, very low jitter and very low packet loss. The targeted quantitative value for end-to-end delay is less than 150msec for 99.99% of the packets, while packet loss is expected to be less than 10^{-6}.

The PVBR network service mainly copes with unresponsive variable bit rate (VBR) sources. Typical candidate applications are real time video and teleconferencing. The requirements are similar to the PCBR network services but with a less strict needs concerning the jitter and packet loss. They are characterized by large packet size, which oscillates from 256 to 1024 bytes. The targeted end-to-end delay is limited to be less than 250msec for 99.99% of the packets, while packet loss should be less than 10^{-4}.

The PMM is expected to carry a mixture of TCP and non-TCP traffic. These flows require a minimum bandwidth, which must be delivered at a high probability. Independently of the transport protocol, flows are expected to implement some kind of congestion control mechanism and their aggressiveness should be similar to the one of TCP, assuming that they are roughly TCP-friendly [3]. This NS is supposed to serve adaptive applications (TCP), like low-quality video, non real time multimedia applications or file transfer (FTP). They require throughput guarantees, which are translated into low packet loss only for “in-profile” packets ($\leq 10^{-3}$).

PMC is targeting to non-greedy adaptive applications that have great sensitivity concerning packet loss. It is thus suitable for transaction-oriented applications and interactive applications such as online games and chat-like applications. The main characteristics are the non-greediness of the flow, the responsive nature (TCP), the low use of bandwidth and the short life of the connection. The most important requirement is very low packet loss only for "in-profile" packets ($\leq 10^{-6}$). Nevertheless, low queuing delay is also desired, in order to retain the meaning of interactiveness.

Finally, packets of the STD BE receive no special treatment in the network.

2.1 Traffic Classes

The implementation of the Network Services is realized with the use of some network's mechanisms, which are the Traffic Classes (TCLs). A TCL is defined as a composition of a set of admission control rules, a set of traffic conditioning rules (Fig.1) and a per-hop behavior (PHB). In the proposed architecture five TCLs are introduced: TCL1, TCL2, TCL3, TCL4 and TCL5 which correspond to PCBR, PVBR, PMM, PMC and STD BE. Each TCL maintains a separate queue at the router output ports and allocates one or more DSCPs in order to enable differentiation of packets in the core network. A PHB implemented in the output port of a router is realized in the network with the use of scheduling and buffer management algorithms. The scheduling mechanism selected is a combination of the Priority Queuing [4] and Weighted-Fair Queuing [4], which is called PQWFQ (Fig.2). TCL1 has a strict priority over the other TCLs. The rest TCLs are scheduled with the WFQ and each queue is managed by different weight and queuing strategy [5].

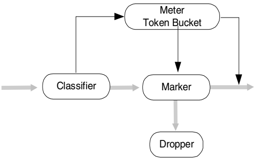


Fig. 1. Traffic Conditioning Mechanisms

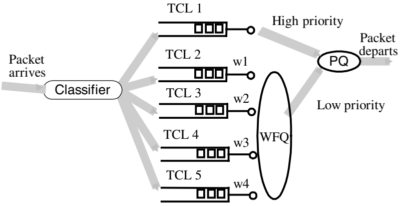


Fig. 2. Design of router output port

According to the WFQ weights, the traffic injected into the network should be limited. Therefore, apart from the traffic classes, specific Admission Control (AC) algorithms should be implemented at the edges of the network to control the admitted number of flows. The proposed AC algorithms for each TCL are described in detail in [6]. Moreover, specific policing actions are deployed to ensure that non-conforming data flows do not affect the QoS requirements for already active data flows. Policing at the network access point is performed through a token bucket (TB) device (r,b). A specific traffic profile is determined for each NS, which best characterizes the data source.

3 Simulations

The simulations were realized in a large-scale network topology. This topology consists of five interconnected networks, which belong to five cities of Europe. Three of them are considered as transit networks, which are situated in Munich, Vienna and Rome. The traffic generators are placed in the network of Athens and the destination network is London for all TCLs, in order to choose the longest path. The routers compromising the end-to-end topology are depicted in Fig.3. Background generators are placed in different links and different domains, rising five different bottlenecks in the network. The EIGRP is considered as the routing protocol for the whole network. The recommended AC limits for each TCL are configured as: 10% for TCL1, 15% for TCL2, 30% for TCL3, 5% for TCL4 and for TCL-STD (BE) is dedicated the rest of the link. Regarding the BT, each TCL is considered with the maximum admitted value of traffic.

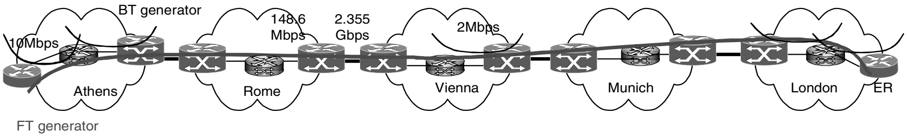


Fig. 3. End-to-End Path

3.1 Study of Tcl1 & Tcl2

TCL1-PCBR is served as foreground traffic using a voice flow. The performance of TCL1 was validated assuming target packet loss ratio to be 10^{-6}.According to the specified AC, the maximum admissible load is ρ=0.52, that is equivalent to 104 kbps. Therefore, a single TB [6] for TCL1 was configured with PR=104kbps and BSP=256Bytes. The buffers in the routers were set to 5 packets for TCL1. The end-to-end delay for different packet sizes was measured without any BT (Fig.4). The end-to-end delay of TCL1 (130Bytes packet size) was also measured for a sequentially increasing number of bottlenecks in the network (Fig.5). The basic conclusion is that increasing the amount of BT the end-to-end delay is being increased up to three times. Although, this value still remains low and less than 150msec.

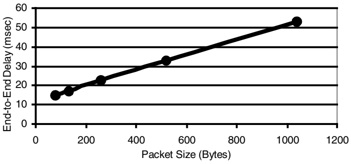


Fig. 4. Av. end-to-end delay vs packet size

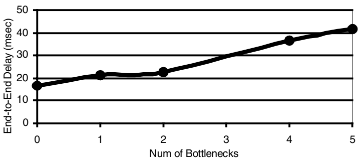


Fig. 5. Av. end-to-end delay vs bottlenecks

TCL2-PVBR class is served as foreground traffic where video flows. Assuming that the AC limit is 300kbps and the target packet loss equal to 10^{-4}, the effective bandwidth for each admitted flow is 34,650kbps, where each flow is characterized by PR=32kbps, SR=24kbps and packet size 400bytes. Therefore, the number of admitted flows is 8. The buffer size in routers for TCL2 was set to 5 packets, in order to avoid long queuing delays. A dual TB [6] was consequently configured for each flow, with PR=32kbps, BSP=1000B(2*M), SR=24kbps and BSS=5000B(10*M). The average end-to-end delay for each flow was measured as depicted in Fig.6, where no BT was used. Moreover, the maximum end-to-end delay was measured having different bottlenecks. These results are depicted in Fig.7. As a final result was that increasing the BT injected in the network, the max. observed end-to-end delay is increased of up to two times; though it still remains less than 250msec.

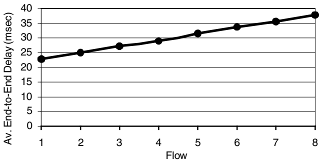


Fig. 6. Av. end-to-end delay vs flow of TCL2

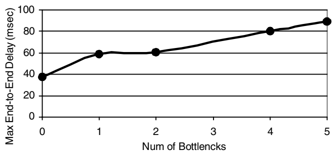


Fig. 7. Max. end-to-end delay vs bottlenecks

3.2 Study of Tcl3 & Tcl4

TCL3-PMM is served as foreground traffic that is targeted for low-quality video and file transfer applications. The dedicated bandwidth for TCL3 was set to 540 kbps, where AC limit was 600kbps and the target utilization factor equal to 0.9. Five TCP flows were used for TCL3 with a mean rate of 108kbps and packet size equal to 1000Bytes. A single TB [6] was configured for each flow with SR=108kbps and BSS=10,000B(10*M). The configuration of the WRED algorithm [3] on a 2Mbit/s link is for “out-profile” packets: min$_{th}$=18, max$_{th}$=38, and 1/maxp=9, for “in-profile” packets: min$_{th}$=38, max$_{th}$=97, and 1/maxp=88. The buffer size was set to 130 packets and the packet loss ratio was considered to be less than 10^{-3}. The results show, that the capacity (600kbps) is shared among these five TCP connections in a fair manner. The total throughput is depicted in Fig.8. This throughput is decreased up to the scheduled bandwidth of TCL3, when a BE traffic is occurred. The measured value of packet loss for “in-profile” packets was $3*10^{-4}$, when the simulation time was 5min.

TCL4-PMC is served as foreground traffic. PMC traffic is simulated through ON-/OFF sources with constantly distributed ON/OFF times. During ON time the source sends TCP packets with an average rate of 23kbps for 2sec with a packet size 500Bytes. The OFF time was set to 2sec. According to the AC limit, the effective bandwidth for each flow is equal to 19.63kbps; so 5 flows will be admitted. A dual TB [6] for TCL4 was configured, with PR=32kbps, BSP=1000B (2*M), SR=14kbps and BSS=5,000B(10*M). A FIFO with two thresholds was considered as the buffer management. The buffer size was set to 35 packets and the dropping threshold for “out-profile” packets to 10 and for “in-profile” packets to 35. The end-to-end delay for TCL4 was measured having different bottlenecks (Fig.9). The measured value of packet loss for the “in-profile” packets was $9*10^{-6}$, when the simulation time was 12h. Consequently, the average end-to-end delay increases while increasing the BT injected in the network, but it still remains low.

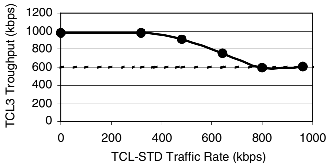


Fig. 8. TCL3 Throughput of five TCP flows

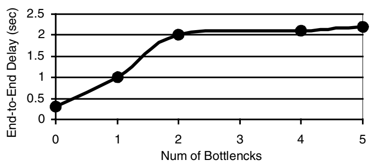


Fig. 9. Average end-to-end delay for TCL4

4 Conclusions & Future Work

The work presented in this paper dealt with the definition and deployment of a set of Network Services within a DiffServ-enabled core network architecture. The Network Services, which are implemented in the network with the traffic handling mechanisms offered by respective Traffic Classes, target at different kinds of user traffic that exhibit similar QoS requirements and characteristics, and they therefore demand analogous treatment within the network. We propose five Network Services that can accommodate most of the well-known application traffic usually submitted in a network. A different set of mechanism is used for each TCL, based on flows characteristics and the corresponding QoS requirements. Subsequently, simulation results proved that the proposed traffic handling mechanisms are adequate for the proposed Network Services, even under the proposed large-scale topology, which compromises a worst-case scenario. Therefore, the correctness of our design was verified, since the target QoS performance was achieved for all the NSs. Future work would focus on refinement of the proposed traffic control mechanisms (traffic conditioner, buffer management, scheduling) and on performance studies using different traffic models.

Acknowledgements. This paper is partly funded by the European research project AQUILA, Information Societies Technology (IST) programme, IST-1999-10077. The authors are currently engaged in the definition, implementation and evaluation of the concept presented in this paper.

References

[1] Black, D.et.al., “An Architecture for Differentiated Services”, RFC 2475, December 1998.

[2] Deliverable D1201, System architecture and specification for the first trial, AQUILA project consortium, http://www-st.inf.tu-dresden.de/aquila/, June 2000.

[3] M. Allman, et.al., “TCP Congestion Control”, RFC 2581.

[4] M. Markaki, E. Nikolouzou, I. Venieris, "Performance Evaluation of Scheduling Algorithms for the Internet", 8th IFIP Conference on Performance Modeling and Evaluation of ATM & IP Networks, Ilkley, June 2000.

[5] T. Ziegler, C. Brandauer, S. Fdida, "A quantitative model for parameter setting of RED with TCP traffic", 9th International Workshop on Quality of Service, Karlsruhe, Germany, June 6-8, 2001.

[6] Deliverable D1301, Specification of traffic handling for the first trial, AQUILA project consortium, http://www-st.inf.tu-dresden.de/aquila/, September 2000.

SaTPEP: A TCP Performance Enhancing Proxy for Satellite Links

Dimitris Velenis, Dimitris Kalogeras, and Basil Maglaris

Department of Electrical and Computer Engineering, Network Management and Optimal Design (NETMODE) Laboratory, National Technical University of Athens,
Heroon Politechniou 9, Zographou, 157 80, Athens, Greece
{dbelen,dkalo,maglaris}@netmode.ece.ntua.gr

Abstract. Satellite link characteristics cause reduced performance in TCP data transfers. In this paper we present SaTPEP, a TCP Performance Enhancing Proxy which attempts to improve TCP performance by performing connection splitting. SaTPEP monitors the satellite link utilization, and assigns to connections window values that reflect the available bandwidth. Loss recovery is based on Negative Acknowledgements. The performance of SaTPEP is investigated in terms of goodput and fairness, through a series of simulation experiments. Results obtained in these experiments, show significant performance improvement in presence of available bandwidth and at high error rates. [1]

1 Introduction

Satellite link characteristics, namely long propagation delays, large $bandwidth \cdot delay$ products, and high bit error rates, affect the performance of TCP, the dominant transport layer protocol in the Internet. In network paths with large $bandwidth \cdot delay$ products, TCP needs a considerable amount of time to set its congestion window, *cwnd*, to the appropriate value [1]. Furthermore, TCP reacts to segment drops by lowering *cwnd* [2]. When drops are caused by transmission errors, TCP unnecessarily reduces its transmission rate.

Several methods to overcome those problems are listed in [3] and [4]. Many of them employ end-to-end mechanisms. Others try to increase performance by mechanisms implemented at certain points in the path between the TCP endpoints [5], [6]. The Satellite Transport Protocol, STP [7], may be used either in a split TCP connection over the satellite part of a network, or as a transport layer protocol within a satellite network. TCP-Peach [8] attempts to improve end-to-end TCP performance in a priority aware environment.

In this paper we introduce SaTPEP, a TCP Performance Enhancing Proxy that aims at increasing the performance of TCP over single-hop satellite links. SaTPEP's flow control is based on link utilization measurements, and segment loss is handled with Negative Acknowledgements (NACKs). The remainder of the paper is organized as follows: In Section 2 we describe the design of SaTPEP. In Section 3 we present simulation results obtained by a SaTPEP model in the ns [9] simulator. Section 4 concludes the paper.

[1] This work is partially supported by OTE S.A. R&D laboratories

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 1233–1238, 2002.

2 Satellite TCP Performance Enhancing Proxy – SaTPEP

SaTPEP consists of the two gateways at each end of a bidirectional satellite link (or a unidirectional satellite forward link and a reverse terestrial link). Every TCP connection traversing the link is split as follows: One connection is established between the TCP sender and the Uplink Gateway (UG), another one between the two gateways, called the *SaTPEP connection*, and a third one between the Downlink Gateway (DG) and the TCP receiver.

In order to improve performance over the satellite hop, the SaTPEP connection performs flow control based on link utilization measurements, and error recovery with Negative Acknowledgements (NACKs).

2.1 Flow Control

A SaTPEP connection begins with the standard TCP three-way handshake. The SaTPEP sender (UG) does not perform any *cwnd* calculations. It just sets *cwnd* to *rwnd*, the window value advertised by the SaTPEP receiver (DG). SaTPEP measures window values in MSS-sized segments, rather than in bytes. At the beginning of a SaTPEP connection, the sender sets *rwnd* to 1. On receipt of the SYN-ACK segment, *rwnd* is set to the value in the window field of the TCP header. This value is calculated by DG as the minimum of the available buffer space for incoming data, and a window value calculated using the link utilization measurement. Given the link capacity, DG can measure its utilization by measuring the incoming data throughput. This throughput measurement is based on the Packet Pair algorithm [10], and it is performed over all received IP traffic. A timer, the *idle timer*, is used to handle periods of link inactivity. When it expires the throughput measurement is set to zero.

Whenever a measurement is completed, DG calculates the available bandwidth by subtracting the throughput measurement from the total bandwidth of the link. The available bandwidth multiplied by the link RTT denotes how much more data the link can attain. We call this value *Available Window*, *AW*. *AW* is distributed to the connections as an increment to their *rwnd* values. DG may use a wide variety of criteria to distribute *AW* to its connections. It might implement policies that favor certain types of traffic, or certain hosts over others.

In the present paper we propose an algorithm for distributing *AW* in a fair manner to all *active* connections. A connection is characterized as active, if it has transmitted at least one data segment during the last throughput measurement. Non-active connections do not get a share from *AW*. Assuming n active connections, the *rwnd* value of the k-th connection is incremented by a value $drwnd_k$, defined in equation (1). Note that the $drwnd_k \cdot MSS_k$ products, of all active connections, sum up to *AW*.

$$drwnd_k = \frac{AW}{MSS_k} \cdot \frac{2 \cdot \sum_{i=1}^{n} rwnd_i - n \cdot rwnd_k}{n \cdot \sum_{i=1}^{n} rwnd_i} \ . \qquad (1)$$

When *AW* is distributed to the active connections, connections with smaller *rwnd* values receive larger *drwnd*, leading to a steady state of fair bandwidth distribution among all active connections. The $rwnd_k$ value is limited by a maximum value, max_rwnd_k, defined in equation (2), where $c \leq 1$, and del the link round-trip propagation delay. Constant c accounts for data that is released from the SaTPEP socket buffer to the IP layer and has not yet been transmitted.

$$max\_rwnd_k = \frac{(1+c)}{n} \cdot \frac{bw \cdot del}{MSS_k} \ . \tag{2}$$

Whenever a connection becomes idle, its *rwnd* is reset to 1 segment. By setting its *cwnd* to *rwnd*, the SaTPEP sender transmits data bursts much larger than a TCP sender. With these bursts transmitted over a one-hop path, a buffer size of at least the $bandwidth \cdot delay$ product of the satellite link is enough to assure that the link will not experience congestion.

2.2 Loss Recovery

The SaTPEP flow control mechanism guaranties that the link will not experience congestion. Therefore, segment drops are only caused by errors, and SaTPEP does not reduce *cwnd* when loss is detected. The SaTPEP sender enters recovery mode when the first duplicate acknowledgement, *dupACK*, is received, since there can be no segment reordering on a single-hop connection. While in recovery mode, *cwnd* is inflated by the amount of dupACKs, *dupwnd*, received. Recovery ends when *recover*, the highest sequence number transmitted when the first dupACK arrived, is acknowledged.

The SaTPEP receiver notifies the sender of missing segments by means of *Negative Acknowledgements*, *NACKs*. NACKs are included in dupACKs as a TCP option, describing a contiguous missing part of the receiver data stream with sequence numbers lower than the maximum sequence number received. The receiver transmits increasingly sequenced NACKs in successive dupACKs, and repeats the same NACKs in a cyclic manner until the data they describe is received.

On receipt of a NACK, the SaTPEP sender retransmits the requested segment(s) along with as much new data as the inflated *cwnd* allows. The sender will not respond to repeated NACKs until a counter, *rtx_count*, expires. *rtx_count* is set to $rwnd - 1$ on receipt of the first dupACK and is decreased by 1 for every dupACK received. It is an estimate of the expected number of dupACKs that will be received before the retransmission reaches the receiver. As long as $rtx\_count > 0$, the retransmitted segment cannot have reached the receiver, and there is no point in repeating the retransmission. When *rtx_count* expires, *rtx_count* is reset and the sender repeats a complete cycle of all retransmissions still requested by incoming NACKs.

SaTPEP also utilizes TCP's Retransmission Timer. Whenever the timer expires the SaTPEP sender sets *rwnd* to 1, *dupwnd* to 0 and retransmits the segment requested by the last ACK received. Figure 1 describes more formally the loss recovery algorithm implemented at the SaTPEP sender.

```
Initially: $dupwnd = 0$, $recover = null$, $hinack = null$, $hiack = null$
           $rtx\_count = null$, $rtx\_allow = 0$, $rtx\_stop = null$
On arrival of 1st dupACK:
  – $hiack \leftarrow$ dupACK's ack_no, $recover \leftarrow$ highest transmitted seq_no, $dupwnd \leftarrow 1$
  – if $rtx\_count$ is $null$ or 0: $rtx\_count \leftarrow rwnd - 1$
  – $hinack \leftarrow$ NACK's highest seq_no
  – retransmit segment(s) requested in NACK option
  – reduce $dupwnd$ by number of segments retransmitted
  – transmit new data (as much as $cwnd$ allows)
On arrival of any dupACK:
  – $dupwnd \leftarrow dupwnd + 1$, $rtx\_count \leftarrow rtx\_count - 1$
  – if NACK's highest seq_no > $hinack$ and $rtx\_allow$=0:
      • retransmit what NACK requests, $hinack \leftarrow$ NACK's highest seq_no
  – else if $rtx\_allow$=1 and NACK does not contain $rtx\_stop$:
      • retransmit segment(s) requested in NACK option
      • if $hinack >$ NACK's highest seq_no: $hinack \leftarrow$ NACK's highest seq_no
  – else if $rtx\_allow$=1 and NACK contains $rtx\_stop$:
      • $rtx\_allow \leftarrow 0$, $rtx\_stop \leftarrow null$, $rtx\_count \leftarrow rwnd$
  – reduce $dupwnd$ by amount of data retransmitted
  – if $rtx\_count = 0$: $rtx\_allow \leftarrow 1$, $rtx\_stop \leftarrow$ NACK's highest seq_no
  – transmit new data (as much as $cwnd$ allows)
On arrival of Partial ACK:
  – reduce $dupwnd$ by Partial ACK's ack_no - $hiack$
  – $hiack \leftarrow$ Partial ACK's ack_no, Perform actions for any dupACK
On arrival of New ACK:
  – $dupwnd \leftarrow 0$, $recover \leftarrow null$, $hiack \leftarrow$ New ACK's ack_no,
    $rtx\_count \leftarrow null$, $rtc\_allow \leftarrow 0$
```

Fig. 1. SaTPEP sender reaction to Duplicate Acknowledgements

3 Simulation Experiments

We evaluate the performance of SaTPEP in comparison to SACK-TCP, in a series of simulation experiments using the *ns* simulator [9]. A bi-directional GEO satellite link is used to establish communication between N data senders and N receivers. The data senders are connected to the Uplink Gateway, and the receivers to the Downlink Gateway. They perform bulk data transfers of various file sizes. The packet size is set to 1500 bytes. The propagation delay of the satellite link is set to $275ms$, resulting in a RTT of $550ms$ between UG and DG. Link capacity values range from 2 to $10Mbps$. All other links have a propagation delay of $1ms$, and their capacity is set to $10Mbps$, or $100Mbps$ in the case of a $10Mbps$ satellite link. The packet loss probability, P_{loss}, of the satellite link, ranges from 10^{-6} to 10^{-2}. All other links are error-free. Queue sizes are set to 600 packets for all links, so that end-to-end TCP transfers do not experience congestion loss. The focus of our comparison is on goodput ($file\ size/connection\ duration$), as perceived by the receiver hosts, and on fairness between multiple simultaneous connections.

In the first series of experiments N is set to 1. All other parameters cover the full ranges already mentioned. In order for TCP to be able to eventually fully utilize the satellite link, we have set the TCP *rwnd* to rather high values, from 100 to 500 segments. Figure 2 depicts goodput achieved by SaTPEP and TCP for different P_{loss} values. The file size is $1Mbyte$ and the link capacity $6Mbps$. SaTPEP performs significantly better than TCP because it is able to fully utilize the link after the first RTT of the connection. Frequent losses cause

TCP's *cwnd* to remain low, while SaTPEP still raises *cwnd* high enough to fully utilize the link. Figure 2 also depicts the goodput ratio for SaTPEP to TCP, which rises significantly for $P_{loss} = 10^{-2}$. For a given P_{loss} value, SaTPEP's performance increases for higher file sizes and link capacities, as shown in figure 3. Both graphs are obtained for $P_{loss} = 10^{-3}$.

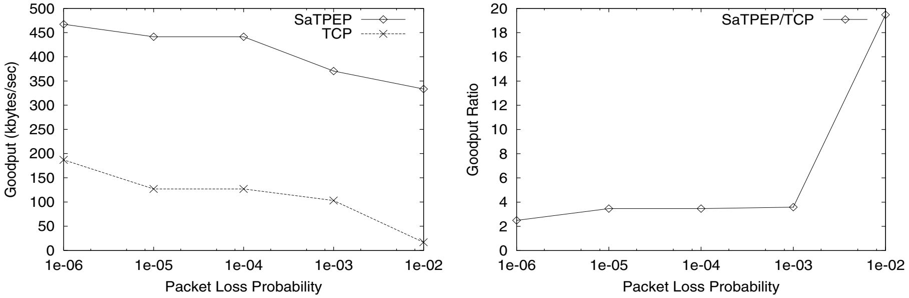


Fig. 2. Goodput and Goodput Ratio for different P_{loss} values

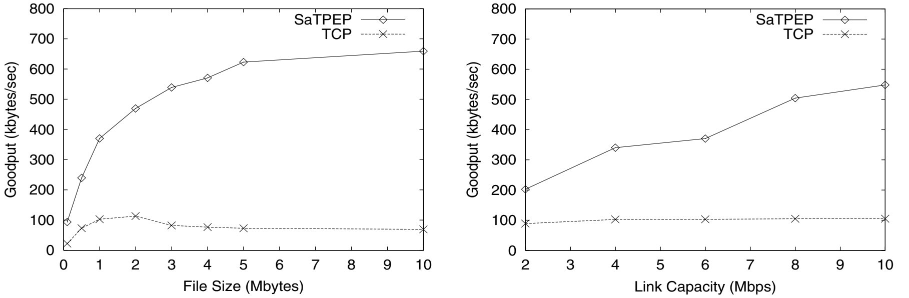


Fig. 3. Goodput for different file size and link capacity values

In the second series of experiments, we set N to 21 and the link capacity to $6Mbps$. At time $t_1 = 1sec$ twenty senders begin transmission of a $2Mbyte$ file each. At time $t_2 = 10sec$ the 21st sender begins transmission of a $500kbyte$ file. The *rwnd* value for TCP connections is set to 25 segments, high enough to result in full utilization of the link, without causing congestion during the initial Slow Start phase. Figure 4 depicts goodput achieved by each of the initial twenty connections for $P_{loss} = 10^{-3}$. It is clear that SaTPEP distributes the link capacity in an even more fair manner than TCP does. The average goodput achieved by the twenty initial connections, along with the goodput of the 21st connection, is shown in figure 4 for different P_{loss} values.

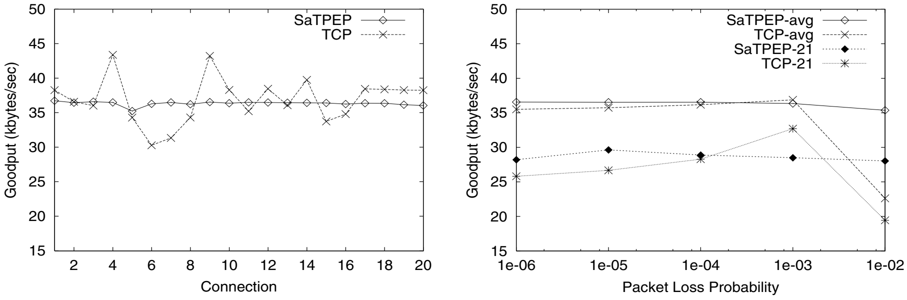


Fig. 4. Goodput of 20 simultaneous connections. Average Goodput of 20 connections, and Goodput of connection 21 for different P_{loss} values

4 Conclusion

In this paper we introduced SaTPEP, aiming at improving TCP performance over satellite links. SaTPEP's flow control is based on link utilization measurements. Loss recovery is based on Negative Acknowledgements. Simulation experiments show significant performance improvement over TCP, in presence of available link capacity, and under high error rates. Under heavy traffic, SaTPEP exhibits remarkable fairness between simultaneous connections.

References

1. C. Partridge and T. Shepard, "TCP/IP Performance over Satellite Links," *IEEE Network Mag.*, pp. 44–49, Sept. 1997.
2. V. Jacobson, "Congestion Avoidance and Control," in *Proc. ACM SIGCOMM*, Stanford, CA USA, Aug. 1988.
3. M. Allman, D. Glover, and L. Sanchez, "Enhancing TCP over Satellite Channels using Standard Mechanisms," RFC 2488, Jan. 1999.
4. M. Allman, S. Dawkins, D. Glover, J. Griner, D. Tran, T. Henderson, J. Heidemann, J. Touch, H. Kruse, S. Ostermann, K. Scott, and J. Semke, "Ongoing TCP Research Related to Satellites," RFC 2760, Aug. 2000.
5. J. Border, M. Kojo, J.Griner, G. Montenegro, and Z. Shelby, "Performance Enhancing Proxies Intended to Mitigate Link-Related Degradations," RFC 3135, June 2001.
6. I. Minei and R. Cohen, "High-Speed Internet Access Through Unidirectional Geostationary Satellite Channels," *IEEE JSAC*, vol. 17, no. 2, pp. 345–359, Feb. 1999.
7. T. Henderson and R. Katz, "Transport Protocols for Internet-Compatible Satellite Networks," *IEEE JSAC*, vol. 17, no. 2, pp. 326–344, Feb. 1999.
8. I. Akyildiz, G. Morabito, and S. Palazzo, "TCP-Peach: A New Congestion Control Scheme for Satellite IP Networks," *IEEE/ACM Transactions on Networking*, vol. 9, no. 3, pp. 307–321, June 2001.
9. "NS (Network Simulator)," http://www.isi.edu/nsnam/ns/.
10. S. Keshav, "A Control-Theoretic Approach to Flow Control," in *Proc. ACM SIGCOMM*, Zurich, Switzerland, Sept. 1991.

An Overlay for Ubiquitous Streaming over Internet

Chai Kiat Yeo[1], Bu Sung Lee[1], and Meng Hwa Er[2]

[1] School of Computer Engineering
[2] School of Electrical & Electronics Engineering
Nanyang Avenue, S639798, Singapore
{Asckyeo, Ebslee, Emher}@ntu.edu.sg

Abstract. Conventional distribution of real-time multimedia data uses multicasting or a series of relays and tunnels for unicast networks. The former is a capability not popularly enabled by a lot of networks while the static relays cannot readily adapt to changing network conditions and are potential bottlenecks in a heavily accessed system. This paper proposes a dynamic overlay framework for streaming multimedia data over heterogeneous networks. The overlay comprises a self-improving tree which is built from client relays on the fly and a lightweight server to manage the tree. The overlay provides a better QoS than conventional relays as it monitors the network and re-configures the tree to adapt to changing environments. Clients can switch parents for better QoS. The robustness of the tree is improved by using a spiral mechanism and failure of the lightweight server will not impact the data distribution functionality of the existing tree.

1 Introduction

The IP multicast [1] has been a highly efficient delivery mechanism for best-effort, large-scale, multi-point delivery of real-time multimedia data. However, Internet Service Providers and organisations deliberately disable multicast traffic to protect their networks against unwanted traffic. With the increasing popularity of multicast and broadband applications, the only way then for intranet clients and multicast-disabled networks to access multicast sessions is through a combination of tunnelling and a network of static relays. [2] and [3] are examples of such applications.

[2] proposes a hierarchical configuration of reflectors to act as unicast-multicast bridges. It uses a clustered-based approach by the manual placement of distributed servers at bottlenecks in the network to balance the load. The problem with this approach is the inability of the system to respond to rapid changes in the network and the potential of these servers becoming bottlenecks themselves.

[3] proposes a centralised framework for developing collaborative applications using a lightweight application level multicast tunnelling called mTunnel [4]. A centralised server is used to view, manage and effect all tunnelled sessions with specific gateways employed to unify unicast-multicast clients. Its drawback is the potential bottleneck in host processing capability and network resources.

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 1239-1244, 2002.

2 Framework Overview and Design

Fig. 1 shows the architecture and the operation of the framework. It comprises the Directory Server (DS), the Web Server (WS) and the overlay tree of client nodes. The overlay tree is responsible for the distribution of data streams while DS is only responsible for the management functions. Hence the load of DS is vastly reduced compared to [3]. WS provides the GUI for sources to advertise their sessions. A separate overlay is built for different sources.

2.1 Overlay Construction and Operation

A source can either be unicasting or multicasting. The former will have to advertise its session by contacting WS (Step 1) while the latter will be automatically discovered by WS via the Session Directory Service (sdr) (Step 1) [5]. The overlay is built using DS as a point of contact [6]. The tree-only approach is much less complex than the tree-mesh approach adopted in [7] and [8]. Note that should DS fail, data distribution will still function normally except that new clients cannot join the tree until DS recovers.

Fig. 2 shows an example of a 4-level overlay tree, rooted at the source. Level 1 clients are multicast-enabled clients (C1 and C2 linked by dotted lines to the source) and proxies (Proxy 1) set up by the framework. The proxies act as relays for unicasting sources as well as a parent for the first unicast client joining the tree. It also doubles up as a static relay in the event of severe client failures. Clients from Level 2 onwards are simply members who join the group over time.

A new member selects the session to join from WS (Step 2 of Fig. 1) and issues a join request to DS (Step 3). DS will search its database and returns a list of potential parents (Step 4) using an algorithm which is similar to Prim's [9], commonly used to derive the minimum spanning tree in multicast routing. The clients are categorised into four groups, i.e. 1 to 4 based on Round Trip Time (RTT) between the client and the source. The categories are derived from data provided by [10, 11]:

Cat 1 RTT < 100 ms Cat 2 100 ms < RTT < 200 ms

Cat 3 200 ms < RTT < 400 ms Cat4 > 400 ms

Cat 1 clients are always chosen to be the parent for a client to ensure that the chosen parent is closest to the originating source. Note that unlike Prim's algorithm, the process does not necessarily mean that the chosen parent is closest to the client. However, the proposed framework is self-improving such that the clients converge towards the closest parent, ultimately reverting to Prim's algorithm again. DS will return a list of 5 parents (where available) in ascending order of categories with a maximum of 3 Cat 1 clients, 1 Cat 2 client and 1 Cat 3 client. The latter two clients are selected randomly. The client will then establish connections with the given Cat 1 potential parents (Step 5) and connect to the parent with the best QoS i.e. closest to it as per Prim's algorithm and update the DS (Step 6).

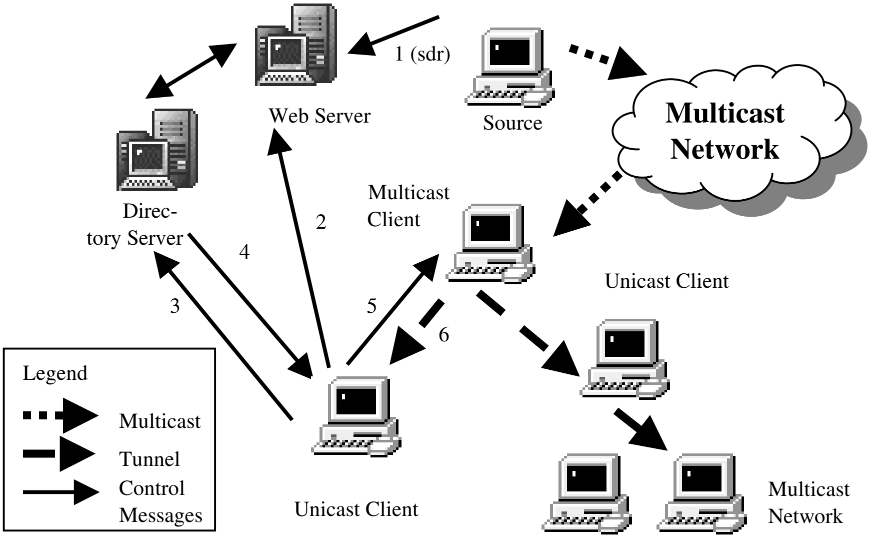


Fig. 1. Overlay Architecture

2.2 Overlay Adaptation

To adapt the overlay to changing network conditions, clients monitor the RTT to their respective parents as well as gossip [12] with the other potential parents returned by DS. Should the RTT results prove to be higher than the initial category of its parent, the client will attempt to switch to a better parent. As illustrated in Fig. 2, C8 gossips with C4, C5 and C6. Note that Cat 2 and Cat 3 nodes are also involved. As the overlay tree strives to improve its quality, the QoS delivered by each client changes. Their inclusion therefore provides a means to avoid partitioning of the tree by having a wider list of gossipers for the client without reverting to DS. Each client sends its own QoS parameter (RTT) to the potential parents that it is gossiping with. If the client finds that the QoS received from other potential parent is better than its current parent, it will perform a parent switch. The client will inform its children about its parent switching so as to avoid an influx of switching among its children.

Switching oscillation is prevented by checking that the QoS history of the potential parent is better than the client's current value by a threshold, and that the client has not switched within a predefined time period, and that the client has not received information that its current parent is also doing a switch, the client can then switch to the new parent.

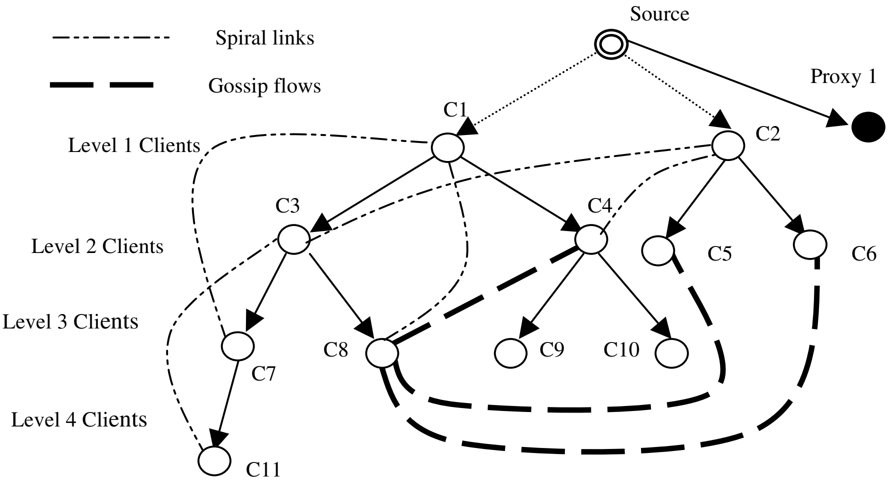


Fig. 2. Example of an Overlay Tree with Spiral and Gossip Mechanisms

2.3 Overlay Robustness

Membership on the overlay tree is dynamic as clients join and leave the tree and experience failures. Spirals shown in Fig. 2 are incorporated to strengthen the tree without incurring the complexity of a full mesh. Spirals can basically withstand node failures in any of its overlay tree branches so long as these failures are not consecutive nodes of the same branch. Client maintains a connection with its grandparent so that should a parent fail, it simply connects to its grandparent without needing to request for a new list of potential parents from DS. Information of the grandparent is passed to the client when it first establishes connection with its parent. Fig. 2 shows spirals from C11 to C3, C7 to C1 which can withstand the failure of clients C7 and C3 respectively. Level 2 clients who do not have grandparents will spiral with the siblings of their parents, e.g. C3 to C2 and C4 to C2.

For consecutive node failures in the same branch, recovery is via the gossip mechanism. If all else fails, the client can simply request DS for a new parent. Client who leaves voluntarily will inform its children, parent, grandparent and grandchildren about its impending departure. The child nodes will then connect to their grandparent (which is the leaving client's parent) immediately. The children who spirals with the leaving client will similarly switch to the leaving client's parent for spiralling.

3 Performance

The framework is implemented in Java using JDKv1.3 and JMF2.1. It has been tested on Win 98/NT and Solaris. Fig. 3 shows the overlay used in the experiments. All the clients are connected via a 100 Base-T switch in a Local Area Network. The clients are Intel P3 500 MHz PCs with 128 MB SDRAM, installed with Win 98 OS. An MPEG2 source with a peak rate of 3.5 Mbps is used. Results are compared to 6 unicast clients sourced by a conventional single static source and the ideal case of 6 multicast-abled clients connected to a multicast source.

3.1 Loss Measurements

The average loss rate per client, shown in Fig. 4, is captured over 10 runs and the experiment is repeated by varying the number of Level 2 and 3 clients from 2 to 4 to 6. Multicast is most efficient for streaming multimedia data regardless of client number with an average loss rate of 0.27%. The static server unicast setup is the least efficient as it cannot scale unlike the overlay tree which scales much better given its distributed nature. The average loss packet per client increases from 0.72% to 1.25% when the number of clients increases from 4 to 6.

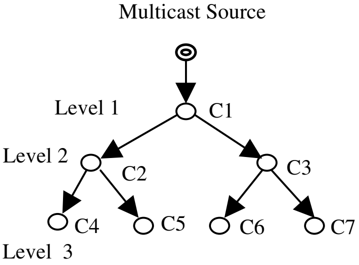


Fig. 3. Overlay used in Experiments

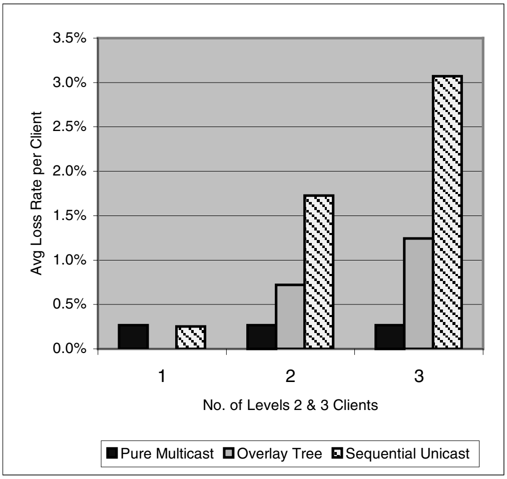


Fig. 4. Average Loss Rate

3.2 Inter-Level Latency

The inter-level latency, shown in Table 1, is measured through a series of ping requests between client and parent as the number of clients varies. The delay is insignificant although it should be noted that it increases with the client number as the response time of the parent gets longer when the parent services more children.

Table 1. Inter-level Latency

# of Level 2 & 3 Clients	2	4	6
Latency (ms)	0.14	0.195	0.32

4 Conclusion

An application level overlay for ubiquitous streaming of multimedia data is proposed. The self-organising and self-improving abilities of the overlay are accomplished through the monitoring of network dynamics. By adapting itself to the prevailing network conditions, better overlays are configured.

References

1. S. E. Deering, Multicast Routing in a Datagram Internetwork, *PhD Thesis*, Stan. U. (1991).
2. M.H. Willebeek-LeMair, Bamba-Audio & Video Streaming over Internet, *IBM J. Res. Develop*, vol. 42, no. 2, Mar (1998) 269-279.
3. Peter Parnes, et al, mSTAR: Enabling Collaborative Applications on the Internet, *IEEE Internet Computing*, Sep-Oct (2000) 32-39.
4. Peter Parnes, et al, Lightweight application level multicast tunnelling using mTunnel, *Computer Communications*, vol. 21 (1998) 1295-1301.
5. Ross Finlayson, Internet Draft: Describing Session Directories in SDP, http://search.ietf.org/internet-drafts/draft-ietf-mmusic-sdp-directory-type-02.txt
6. M. Kadansky, et. al., Reliable Multicast Transport Building Block: Tree Auto-Configuration, IETF RMT WG, draft-ieft-rmt-bb-tree-config-02.txt, 2 Mar (2001).
7. Y.H. Chu, S.G. Rao, S. Seshan, H. Zhang, Enabling Conferencing Applications on the Internet using an Overlay Multicast Arch., SIGCOMM, San Diego, California, Aug (2001).
8. P. Francis, Yoid: Extending the Internet Multicast Arch. ,http://www.aciri.org/yoid (2000).
9. Prim, R. C., Shortest connection networks and some generalizations, Bell Sys. Tech Journal, 36, (1957), 1389-1401.
10. T. Hansen, J. Otero, T. McGregor, H.W. Braun, Active Measurement Data Analysis Techniques, Int. Conf. On Communications in Computing, Las Vegas, Jun (2000) 105-135.
11. Internet Traffic Report, http://www.internettrafficreport.com/index.html.
12. Q. Sun, Sturman, D.C., A gossip-based reliable multicast for large-scale high-throughput applications, *Proc. Conf .Dependable Systems & Networks* (2000) 347 -358.

A Measurement-Based Dynamic Guard Channel Scheme for Handover Prioritization in Cellular Networks

Roland Zander and Johan M. Karlsson

Department of Communication Systems, Lund University
Box 118, SE-221 00 Lund, Sweden
{rolandz, johan}@telecom.lth.se

Abstract. The introduction of guard channels in a cellular network is a method for giving priority to on-going calls by having channels exclusively reserved for handover purposes. Herein, an adaptive measurement-based dynamic guard channel scheme is introduced. The proposed scheme uses the number of on-going calls in adjacent cells and measurements of handover probabilities to determine the amount of guard channels to allocate in a cell. To improve the efficiency of the scheme, the calls are divided into groups depending upon mobility and latest visited cell, where separate measurements are performed for every single group. Simulations showed that the proposed scheme seems to be very efficient.

1 Introduction

The quality of service (QoS) in a cellular network consists among other things of the blocking probability for a new call due to channel occupancy and the probability for a forced call termination. An on-going call may be terminated prematurely due to a handover attempt failure or because of low signal to noise ratio and/or high attenuation. Obviously, it is much more frustrating for a subscriber to have an on-going call dropped than a new call blocked. Since the operator would like to keep the subscribers satisfied, it is a good idea to lower the probability for a forced call termination, which can be achieved by insertion of guard channels. Guard channels are channels exclusively reserved for handover calls, lowering the handover dropping probabilities to the expense of higher blocking probabilities and in most cases reduced throughput.

In the fixed guard channel scheme, a fixed number of guard channels, N_g, is allocated in a cell [1]. A new call is accepted if at least $N_g + 1$ channels are available in the cell at the time of the call arrival, while a handover call only needs one available channel to be accepted. The fixed guard channel scheme is quite simple to implement but unfortunately it is not especially efficient due to its poor flexibility. In dynamic guard channel schemes, the number of guard channels allocated in a cell is time varying and dependent upon the momentary network conditions [2,3]. A well-designed dynamic guard channel scheme provides a higher QoS than its fixed counterpart, but the most important advantage

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 1245–1251, 2002.

with the dynamic schemes is the increased flexibility, making it possible to provide a certain average QoS to on-going calls. Unfortunately, the complexity of dynamic guard channel schemes is higher, requiring increased communication between base stations and increased processing load.

In this paper, an adaptive measurement-based dynamic guard channel scheme is introduced. To improve the efficiency of the proposed scheme, the calls are divided into groups depending upon mobility and most recent source cell (latest visited cell), where separate measurements are performed for every single group.

2 The Proposed Algorithm

The proposed dynamic guard channel scheme uses measurements to estimate the momentary handover arrival intensity of every single cell. From these intensities, the number of guard channels to be allocated in the cells are derived. All base stations measure the probability for an on-going call residing in its coverage area to make a handover attempt (handover probability) and the probability for a handover attempt to take place to a specific target cell (handover direction probability). To improve the accuracy of the estimations, the calls are divided into groups depending upon most recent source cell, where separate measurements are performed for every single group [4,5]. The introduction of source cell groups is very effective since calls belonging to different groups usually do not have the same movement or mobility patterns, which is especially obvious for a call covering a highway where almost all subscribers follow the same path, namely the road. Obviously, the handover direction probabilities for subscribers traveling in opposite directions on the road differ completely from one another.

The handover probability for calls belonging to source cell group x is denoted $P_h(x)$ and the handover direction probability to target cell y for those calls is denoted $P_{hd}(x, y)$. In formulas (1) and (2), ζ is the number of adjacent cells to the cell for which the measurements are performed. $H(x, y)$ is the number of handover attempts to adjacent cell y made by calls belonging to source cell group x, while $D(x)$ is the number of calls belonging to source cell group x having departed from the cell either through a handover or a call termination.

$$P_h(x) = \frac{\sum_{k=1}^{\zeta} H(x, k)}{D(x)} \tag{1}$$

$$P_{hd}(x, y) = \frac{H(x, y)}{\sum_{k=1}^{\zeta} H(x, k)} \tag{2}$$

In order to make the proposed scheme sensitive towards changes in subscriber behavior, only a limited amount of data is taken into account in the measurements. A data value is considered by the scheme if it is less than t minutes old or if it belongs to the z most recent data values. These parameters have to be set as a compromise between adaptability and accurate measurements. To be able to handle more short-term changes in subscriber behavior, recent data are given a larger influence, which is achieved by weighting the data according to a function with a negative exponential behavior.

Most subscribers are stationary while using their mobiles. Consequently, a large portion of the on-going calls is not moving and will accordingly never perform a handover. When predicting subscriber movements it would be useful if the non-moving calls could be distinguished from the moving calls. By letting the covering base station sample the received signal strength from a mobile and calculate the mean value of the last v samples, a signal strength alteration indicating a changed distance between mobile and base station can be detected. In the proposed scheme, a new call is initially placed in a mobility group for undecided calls. If a call movement is detected, the call is transferred to the moving mobility group and the signal strength measuring stops. Calls belonging to the undecided group are after w seconds transferred to the non-moving mobility group. In this case, the measuring continues and if a call movement eventually is detected, the call is transferred to the moving mobility group.

Obviously, all calls arriving from adjacent cells are moving, which makes the moving and non-moving mobility groups unnecessary. Instead, these calls are divided into low and high mobility groups depending upon channel holding time in the latest visited cell. The channel holding time is defined as the time duration between the time a channel is occupied by a call and the time it is released either through a call termination or a handover. If the channel holding time of a call is larger/smaller than the mean value of calls from the same source cell group with identical target cell, the call is placed in the low/high mobility group.

Separate measurements of P_h and P_{hd} are performed for all mobility groups belonging to a certain source cell group. From these probabilities and the number of on-going calls in each group, $C(k)$, the number of calls in a cell expected to make a handover attempt to adjacent cell y, $G(y)$, is derived. There exist $\zeta + 1$ different source cell groups, one for every adjacent cell plus one group for calls with no previous handover. These source cell groups consist of two (low/high) and three mobility groups (moving/non-moving/undecided) respectively. Thus, $2\zeta + 3$ separate measurements are performed for every cell.

$$G(y) = \sum_{k=1}^{2\zeta+3} C(k)P_h(k)P_{hd}(k, y) \tag{3}$$

$G(y)$ is signaled to the base station covering cell y where all received handover expectancy numbers, are summed up. In formula (4), $G_i(y)$ is the number received from adjacent cell i. The resulting total handover expectancy number gives an indication of the current handover arrival intensity and can therefore be used to determine the number of guard channels to allocate in the cell. Since all handover calls will not arrive simultaneously, the actual number of guard channels, N_g, is significantly smaller than the total handover expectancy number. In the proposed scheme, N_g is equal to the total handover expectancy number multiplied by ψ ($\psi < 1$). Fractional guard channels are used, which means that the number of guard channels in a cell does not have to be an integer.

$$N_g = \psi \sum_{i=1}^{\zeta} G_i(y) \tag{4}$$

In order to provide a guaranteed average QoS to already accepted calls, a requested mean value for the handover dropping probability, P_{req}, is set for every cell. By letting ψ be time varying, the handover dropping probability can be held at the requested value. Every single time an on-going call is dropped due to a handover failure, ψ is multiplied by $1 + \kappa$ ($0 < \kappa < 1$), which increases the number of allocated guard channels. When $(1/P_{req}) - 1$ handover attempts have succeeded, ψ is multiplied by $1 - \kappa$ and the counter is set to zero. κ has to be set as a compromise between having a robust scheme (small value) and a scheme sensitive towards traffic alterations (large value).

3 Simulation Model and Numerical Results

The simulated network consisted of 100 rectangular-shaped cells covering a rectangular street network. All cells had four adjacent cells and were arranged in a 10*10 ring topology with wrapped around edges. Hence, a cell in the 1^{st} row with coordinates (x,1) was neighbor with a cell in the 10^{th} row, coordinates (x,10). A Manhattan cell architecture was used, meaning that a base station was placed in every intersection and the cell borders were located midway between adjacent intersections.

All cells in the simulated network were identical from a traffic parameter point of view. The number of channels in each cell were set to 50 and the time duration between consecutive new call arrivals and the call lengths were assumed to be exponential distributed with mean values 0.13 and 5 minutes, respectively. The traffic parameter values were chosen to obtain realistic simulation settings and reasonable traffic load. Three different kinds of users were used in the simulations; stationary, slow moving and fast moving. 50 percent of the users were stationary, 17 percent slow moving and 33 percent fast moving. The channel holding time distribution was either exponential or rectangular (uniform) distributed with mean values of 1 minute (fast moving) and 5 minutes (slow moving).

Six variants of the proposed guard channel schemes, briefly described in Table 1, were investigated. Due to simplification reasons, it was assumed that a new call instantly is placed in a moving or non-moving mobility group. This is performed without errors in scheme VI, while 10 percent of the slow moving calls are placed in the non-moving groups in schemes IV and V. In addition, calls with at least one previous handover are placed in a low or high mobility group in schemes V and VI.

Three simulation scenarios with different channel holding time distributions and handover direction probabilities were investigated. In the handover direction probability sets shown in Table 2, P_1 is the probability for a user to go

Table 1. Investigated guard channel schemes

Scheme	*Description*
I	Fixed guard channels
II	No source cell or mobility groups
III	Source cell groups
IV	Source cell and moving/non-moving mobility groups, inserted errors
V	Source cell and full mobility groups, inserted errors
VI	Source cell and full mobility groups, no inserted errors

straight ahead at an intersection, P_2 the probability for a right turn and P_3 the probability for a left turn.

Table 2. Simulation scenarios

Scenario	*Distribution*	*Handover direction probabilities*
I	Exponential	P_1=0,5 , P_2=0,25 , P_3=0,25
II	Exponential	P_1=0,67 , P_2=0,33 , P_3=0
III	Rectangular	P_1=0,5 , P_2=0,25 , P_3=0,25

In order to shorten the simulation length, it was decided to perform the simulations without the use of the guaranteed average QoS feature. Instead, the handover dropping probabilities were set to $3 * 10^{-4}$ by manual calibration of ψ. This is somewhat unfair towards the fixed guard channel scheme, but this scheme can anyhow be discarded out of flexibility reasons. The blocking probabilities, P_b, were used to determine the best scheme in each specific case. In Table 3, 95 percent confidence intervals are given for the blocking probabilities. All schemes were compared to the fixed guard channel scheme, and the *Gain* column, which shows the obtained gain in percent, is calculated from the two closest and two most distant values in the blocking probabilities of the respective schemes.

Table 3. Simulation results

Scheme	*Scenario I*		*Scenario II*		*Scenario III*	
	P_b(%)	*Gain*	*P_b(%)*	*Gain*	*P_b(%)*	*Gain*
I	3,689-3,693	0	3,690-3,694	0	3,898-3,902	0
II	3,646-3,650	1,1-1,3	3,643-3,648	1,1-1,4	3,835-3,839	1,5-1,7
III	3,594-3,598	2,5-2,7	3,541-3,546	3,9-4,1	3,747-3,753	3,7-4,0
IV	3,574-3,579	3,0-3,2	3,520-3,525	4,5-4,7	3,711-3,716	4,7-4,9
V	3,573-3,578	3,0-3,2	3,517-3,522	4,6-4,8	3,678-3,684	5,5-5,7
VI	3,574-3,578	3,0-3,2	3,520-3,524	4,5-4,7	3,675-3,680	5,6-5,8

The gain obtained by dividing the calls into groups depending upon most recent source cell (scheme III-VI) increased with a larger difference in user behavior between calls from different groups, which can be seen when comparing the simulation results of scenarios I and II. The use of non-moving and moving mobility groups (scheme IV-VI) also led to significant improvements. It was found that the use of high and low mobility groups (scheme V-VI) only was effective for scenario III. In scenarios I and II where an exponential distribution was used, a call with a very long channel holding time in the latest visited cell (low mobility) may out of randomization reasons have a really small channel holding time in the next cell and vice versa. Obviously, this reduces the obtained gain from the prediction-oriented mobility group feature. The removal of the inserted error in the mobility classification procedure (scheme VI) did not have a significant impact on the results.

In general, a large standard deviation for the channel holding time distribution reduces the gain obtained from predictive-oriented features such as source cell and mobility groups because of the larger probability of getting a really small sample value [6]. If a call shortly after arrival in a cell makes a handover attempt, the allocation of guard channels in the adjacent cells may due to channel occupancy not have been fully activated.

4 Conclusions

In this paper, an adaptive measurement-based dynamic guard channel scheme was proposed. The scheme uses the number of on-going calls in adjacent cells and their handover probabilities to estimate the handover arrival intensities of every single cell. In order to improve the accuracy of the estimations, the calls are divided into measurement groups depending upon most recent source cell and mobility, where separate measurements are performed for every single group. The channel holding time in the latest visited cell is used to divide the calls into a low or high mobility group, while positioning is used to divide calls with no previous handover into a moving or non-moving mobility group.

The proposed scheme was compared to other similar guard channel schemes and showed better results (lower blocking probabilities) for all investigated simulation scenarios. However, the high complexity of the scheme requires increased communication between base stations and increased processing load, which has not been taken into account.

References

1. D. Hong and S. S. Rappaport. *Traffic Model and Performance Analysis for Cellular Mobile Radio Telephone Systems with Prioritized and Nonprioritized Handoff Procedures.* IEEE Transactions on Vehicular Technology, vol. 35, no. 3. pp. 77-92, 1986.

2. K. C. Chua, B. Bensaou, W. Zhuang and S. Y. Choo. *Dynamic Channel Reservation (DCR) Scheme for Handoff Prioritization in Mobile Micro/Picocellular Networks.* IEEE ICUPC '98, vol. 1, pp. 383-387, 1998.
3. C. Oliveira, J. B. Kim and T. Suda. *An Adaptive Bandwidth Reservation Scheme for High-Speed Multimedia Wireless Networks.* IEEE Journal on Selected Areas in Communications, vol. 16, pp. 858-874, 1998.
4. C. H. Choi, M. I. Kim, T. J. Kim and S. J. Kim. *Adaptive Bandwidth Reservation Mechanism using Mobility Probability in Mobile Multimedia Computing Environment.* IEEE Local Computer Networks 2000, pp. 76-85, 2000.
5. S. Choi and K. G. Shin. *Predictive and Adaptive Bandwidth Reservation for Handoffs in QoS-Sensitive Cellular Networks.* ACM SIGCOMM '98, pp. 155-166, 1998.
6. R. Zander and J. M. Karlsson. *An Adaptive Algorithm for Allocation of Dynamic Guard Channels -Impact of the Channel Holding Time Distribution.* Wireless 2001, pp. 300-308, 2001.

Author Index