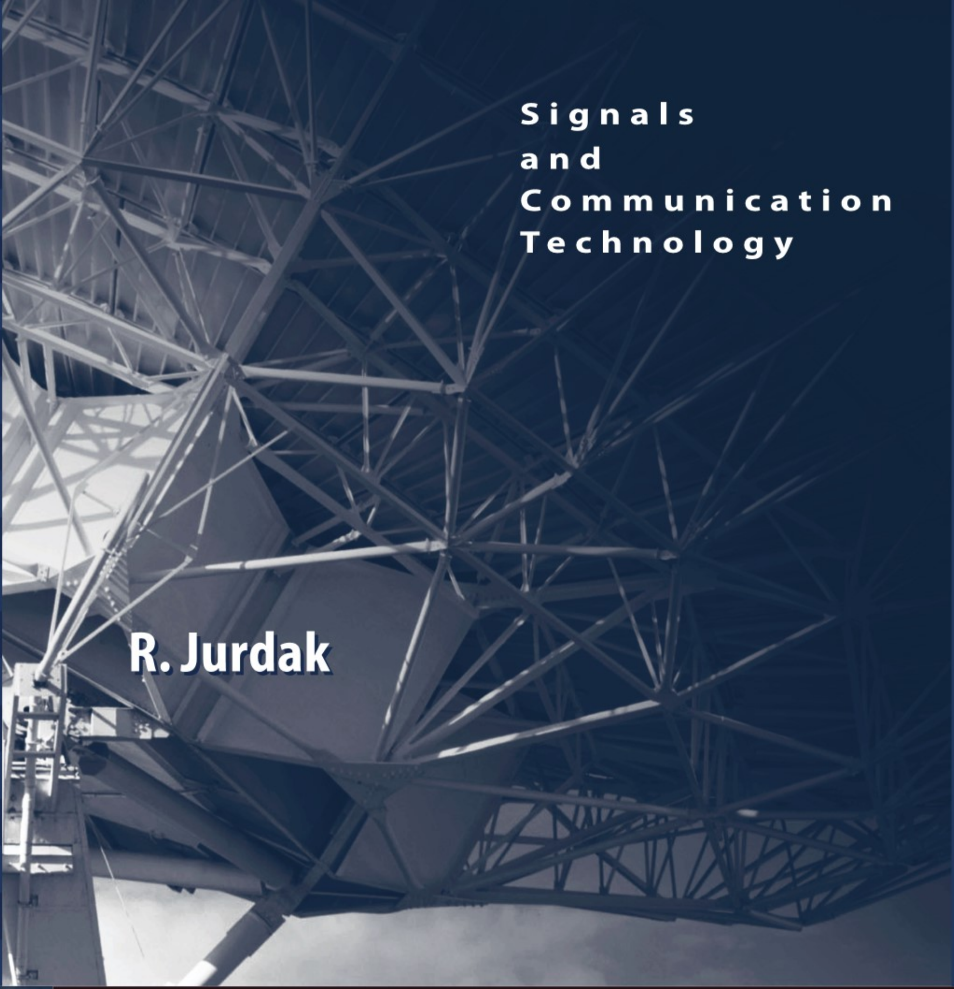


Signals
and
Communication
Technology

R. Jurdak

Wireless Ad Hoc and Sensor Networks

A Cross-Layer Design Perspective

Springer

Springer Series on
SIGNALS AND COMMUNICATION TECHNOLOGY

SIGNALS AND COMMUNICATION TECHNOLOGY

Wireless Ad Hoc and Sensor Networks
A Cross-Layer Design Perspective
R. Jurdak
ISBN 0-387-39022-7

Cryptographic Algorithms on Reconfigurable Hardware
F. Rodriguez-Henriquez, N.A. Saqib, A. Díaz Pérez, and C.K. Koc
ISBN 0-387-33956-6

Multimedia Database Retrieval
A Human-Centered Approach
P. Muneesawang and L. Guan
ISBN 0-387-25627-X

Broadband Fixed Wireless Access
A System Perspective
M. Engels and F. Petre
ISBN 0-387-33956-6

Distributed Cooperative Laboratories
Networking, Instrumentation, and Measurements
F. Davoli, S. Palazzo and S. Zappatore (Eds.)
ISBN 0-387-29811-8

The Variational Bayes Method in Signal Processing
V. Šmídl and A. Quinn
ISBN 3-540-28819-8

Topics in Acoustic Echo and Noise Control
Selected Methods for the Cancellation of Acoustical Echoes, the Reduction of Background Noise, and Speech Processing
E. Hänsler and G. Schmidt (Eds.)
ISBN 3-540-33212-x

EM Modeling of Antennas and RF Components for Wireless Communication Systems
F. Gustrau, D. Manteuffel
ISBN 3-540-28614-4

Interactive Video
Methods and Applications
R. I Hammoud (Ed.)
ISBN 3-540-33214-6

ContinuousTime Signals
Y. Shmaliy
ISBN 1-4020-4817-3

Voice and Speech Quality Perception
Assessment and Evaluation
U. Jekosch
ISBN 3-540-24095-0

Advanced ManMachine Interaction
Fundamentals and Implementation

K.-F. Kraiss
ISBN 3-540-30618-8

Orthogonal Frequency Division Multiplexing for Wireless Communications
Y. (Geoffrey) Li and G.L. Stüber (Eds.)
ISBN 0-387-29095-8

Circuits and Systems
Based on Delta Modulation
Linear, Nonlinear and Mixed Mode Processing
D.G. Zrilic ISBN 3-540-23751-8

Functional Structures in Networks
AMLn—A Language for Model Driven Development of Telecom Systems
T. Muth ISBN 3-540-22545-5

RadioWave Propagation for Telecommunication Applications
H. Sizun ISBN 3-540-40758-8

Electronic Noise and Interfering Signals
Principles and Applications
G. Vasilescu ISBN 3-540-40741-3

DVB
The Family of International Standards for Digital Video Broadcasting, 2nd ed.
U. Reimers ISBN 3-540-43545-X

Digital Interactive TV and Metadata
Future Broadcast Multimedia
A. Lugmayr, S. Niiranen, and S. Kalli
ISBN 3-387-20843-7

Adaptive Antenna Arrays
Trends and Applications
S. Chandran (Ed.) ISBN 3-540-20199-8

Digital Signal Processing with Field Programmable Gate Arrays
U. Meyer-Baese ISBN 3-540-21119-5

Neuro-Fuzzy and Fuzzy Neural Applications in Telecommunications
P. Stavroulakis (Ed.) ISBN 3-540-40759-6

SDMA for Multipath Wireless Channels
Limiting Characteristics and Stochastic Models
I.P. Kovalyov ISBN 3-540-40225-X

Digital Television
A Practical Guide for Engineers
W. Fischer ISBN 3-540-01155-2

continued after index

Raja Jurdak

Wireless Ad Hoc and Sensor Networks

A Cross-Layer Design Perspective

Springer

Raja Jurdak
University College Dublin
Dublín, Ireland

Wireless Ad Hoc and Sensor Networks: A Cross-Layer Design Perspective

Library of Congress Control Number: 2006935264

ISBN 978-0-387-39022-2 e-ISBN 978-0-387-39023-9
ISBN 0-387-39022-7 e-ISBN 0-387-39023-5

Printed on acid-free paper.

Printed in the United States of America.

9 8 7 6 5 4 3 2 1

springer.com

To
Murad, Muna, and Hania
Whose input and support have greatly improved the structure and presentation of ideas in this book.

Preface

Wireless Ad Hoc and Sensor Networks: A Cross-Layer Design Perspective deals with the emerging design trend that transcends traditional communication layers for performance gains in ad hoc and sensor networks.

Recent technological advances have fueled research in the fields of ad hoc and sensor networks that have applications in military, environmental, medical, and civilian domains. Alongside the novel opportunities, the distributed infrastructureless nature of ad hoc and sensor networks poses new challenges for network designers, such as the distribution of network management across resource-limited nodes. To meet the unique challenges of ad hoc and sensor networks and to efficiently utilize the limited node resources, researchers have proposed novel approaches and architectures that implicitly and explicitly violate strictly layered design, cutting across traditional layer boundaries.

Since a comprehensive resource on ad hoc and sensor network cross-layer design is not yet available, this book attempts to fill the gap through a structured comparison and analysis of both layered and cross-layer design. The book also provides 3 case studies for illustrating the benefits of cross-layer design. The book is written with the goal of providing students and researchers with comprehensive overviews on the issues relating to cross-layer design in ad hoc and sensor networks, offering numerous references.

Due to its interdisciplinary character, the book is bound to attract readers from many different areas, such as software engineers, hardware engineers, application developers, network protocol designers, graduate students, communication engineers, systems engineers, and university professors.

The author would like to acknowledge the contributions and support of Cristina Videira Lopes and Pierre Baldi in developing some of the concepts in this book, particularly the case studies.

Contents

Part I Layered Communication Approaches

Part II Cross-Layer Approaches

Part III Case Studies

1

Ad Hoc and Sensor Networks: Opportunities and Challenges

The main goal of wireless ad hoc networks is to allow a group of communication nodes to set up and maintain a network among themselves, without the support of a base station or a central controller. From the applications perspective, wireless ad hoc networks are useful for situations that require quick or infrastructureless local network deployment, such as crisis response, conference meetings, military applications, and possibly home and office networks. Ad hoc networks could, for instance, empower medical personnel and civil servants to better coordinate their efforts during large-scale emergencies that bring infrastructure networks down, such as the September 11 attacks or the 2003 blackout in the northeast region of the United States.

An important subclass of ad hoc networks is wireless sensor networks. The central premise of sensor networks is the distributed collection and digitization of data from a physical space, providing an interface between the physical and digital domains. Sensor networks consist of a potentially large number of sensor modules that integrate memory, communication, processing, and sensing capabilities. The sensor modules form ad hoc networks in order to share the collected physical data and to provide this data to the network user or operator. Sensor networks have a wide range of applications, including medical, environmental, military, industrial, and commercial applications.

Along with the application opportunities of ad hoc and sensor networks, new challenges emerge. The lack of infrastructure in ad hoc and sensor networks requires the nodes to perform the network setup, management and control among themselves. Each node must act as a router and data forwarder in addition to playing the role of a data terminal. Distributing network management across the nodes places a burden on the resources of individual nodes. This additional load at each node complicates the protocol design and performance optimization of ad hoc and sensor networks.

Traditionally, network design has followed a layered communication architecture in which protocols at each layer of the stack handle specific network functions. By providing standardized interfaces between neighboring communication layers in a stack, layered architectures provide a high degree of modularity and interoperability among heterogeneous networks. The most prominent layered model is the 7-layer Open System Interconnection (OSI) reference model [1], proposed by the International Standards Organization (ISO). Two widely implemented layered architectures are the Internet TCP/IP model [2] and the Global System for Mobile Communication (GSM)-based model [3] for cellular phone networks. The OSI reference model and similar strictly layered models are designed for conventional wired and wireless networks with infrastructure that is responsible for network management. Nodes in most traditional networks use their resources only for data communications, while the infrastructure runs centralized algorithms in determining optimal network behavior. In contrast, ad hoc and sensor network node resources must support network formation and management activities, in addition to data communication. The lack of infrastructure in ad hoc and sensor networks often dictates the uses of distributed algorithm, representing an additional level of design complexity. Thus, the optimization of resource usage and optimization in ad hoc and sensor networks is more critical. The importance of the performance optimization factor in ad hoc and sensor networks warrants a reexamination of strictly layered communication architectures.

Layered communication architectures trade off generality for efficiency, because the transparency of one layer to other layers ensures modularity while preventing interlayer cooperation in optimizing network behavior. For example, energy efficiency is a major design goal for ad hoc and sensor networks, given the ever-decreasing form factor of the nodes. The energy consumption of a node is an aspect that transcends traditional layers. The medium access strategy contributes to energy consumption, for instance through collisions. Routing and transport layers strategies and control messages also affect energy consumption. Enabling the medium access, routing, and transport layers to cooperate can promote energy-efficient behavior in the network. This example highlights the need for fine-grained optimizations in ad hoc and sensor networks based on interlayer cooperation, which has led to the proposal of cross-layer design for improving network performance. Cross-layer design emphasizes performance optimization by enabling different layers of the communication stack to share state information or to coordinate their actions in order to jointly optimize network performance. For example, supplying information on a node's remaining battery energy to all of the nodes' communication layers can enable each layer to adjust its configuration for energy-efficient behavior.

This book aims at exploring the current state of the art in cross-layer approaches for ad hoc and sensor networks. In order to provide a fair and comprehensive view, the first part of the book focuses on layered approaches and their applicability to ad hoc and sensor networks. In particular, Part I of the book adopts the OSI model as the reference architecture, and each chapter in Part I presents the issues involved at a particular layer within the OSI model, in an attempt to reveal the opportunities of cross-layer enhancements. Chapter 2 discusses the physical layer aspects, including communication media and technologies of ad hoc and sensor networks. Chapter 3 explores data link layer issues, focusing on the Medium Access Control (MAC) and the logical link control (LLC) sublayers. Chapter 4 provides an overview of the routing considerations in ad hoc and sensor networks. Chapter 5 explores the functions of the communication layers above the routing layer, including the transport, session and presentation layers. Chapter 6 concludes Part I with a discussion of application examples and issues for ad hoc and sensor networks.

The second part of the book focuses on cross-layer design. Chapter 7 explores the issues to consider for cross-layer design in ad hoc and sensor networks. Chapter 8 presents the proposed cross-layer architectures for ad hoc and sensor networks. To apply the architectural concepts of Chapters 7 and 8, Chapter 9 surveys and compares the applied cross-layer approaches for ad hoc and sensor networks, including the author's own cross-layer framework for optimizing these networks. This framework serves as the tool for showing the benefits of cross-layer approaches for ad hoc and sensor networks through three diverse case studies, which constitute the third part of the book. Chapter 10 presents a case study of a monitoring sensor network using Radio Frequency (RF) waves. The case study uses the cross-layer framework to significantly reduce energy consumption in the network and to validate the benefits through deployment experiments. Chapter 11 customizes the framework for an ad hoc network that uses Ultra Wide Band (UWB) radio in order to maximize throughput, promote fairness, reduce latency, and reduce control overhead. The final case study in Chapter 12 discusses an acoustic underwater sensor network for environmental monitoring. In this case, the framework helps prolong the network lifetime through cross-layer optimizations based on topology and transmission frequency.

Part I

Layered Communication Approaches

2

Physical Layer

r The first layer in the OSI reference model is the physical layer, as shown in Fig. 2.1. The physical layer specifies the communication media, the type of energy used for communication, and the mapping of information bits to energy. A transmitter can send a signal through a variety of physical media, including wires, air, and water. Several communication technologies that use different energy types for encoding information also exist, such as radio frequency waves, microwave, infrared, ultra wide band radio, and acoustics. In this chapter, we briefly explore the communication media and technology possibilities for ad hoc and sensor networks.

The chapter is structured as follows. Section 2.1 surveys the types of communication media. Section 2.2 discusses the communication technology alternative for ad hoc and sensor networks.

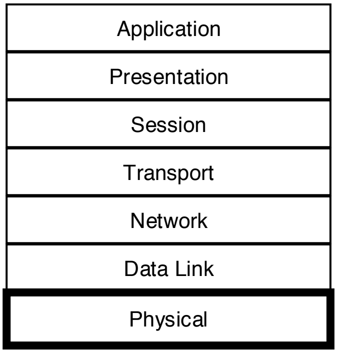


Fig. 2.1. The physical layer in the OSI reference model

2.1 Communication Media

The communication medium specifies the physical channel over which signals are transmitted. Communication media fall into 2 broad categories: wired communications and wireless communications. The remainder of this section explores each category in more detail.

2.1.1 Wired Communication

Wired communications involve signal transmission over a wire or cable. Transmitting signals over wires provides a high degree of control over the signal path, so the quality of signals in wired communications is more stable and relatively higher than the signal quality of comparable wireless communications.

A common type of wire used for both telephone communication and wired local area networks is the copper twisted pair. Twisted pair cables consist of two conducting wires wound around each other in order to reduce electromagnetic interference, referred to as crosstalk, and a plastic enclosure. There are several categories of twisted pair cables that provide different degrees of signal quality depending on the number of twists per meter and the shielding type. For example, category 3 twisted pair cables typically support lower bit rate voice communication on telephone networks, in the range of Kbits per second. Category 5 cables support higher speed data transfer for local area networks, up to 100 Mbits per second. Shielded versions of category 5 and the higher grade category 6 cables can support even higher bit transfer rates, making them suitable for Gigabit ethernet networks.

Another type of wire that also supports local area networks is the coaxial cable, which is used as a transmission line to carry a high-frequency or broadband signal. Coaxial cables consist of a round conducting wire, surrounded by an insulating spacer, surrounded by a cylindrical conducting sheath, usually surrounded by a final insulating layer. The magnetic field created between the conducting sheath and conducting wire is used for transmitting broadband signals, including cable television signals. Coaxial cable is attractive for its relative immunity to outside interference sources, yielding a high signal quality.

The final type of wired communication media we consider is the fibre optic cable. These cables promise extremely high bit transfer rates because of the huge available bandwidth. Fiber optic cables typically serve as long haul communication lines for transcontinental communications as well as shorter range high speed communications.

Wired communication media can serve certain applications of ad hoc and sensor networks. For example, a set of laptops can use category 5 cables to autonomously form an ad hoc network. Similarly, a network of sensors connected by wires on the ceiling of a factory can keep track of merchandize movements. However, wired communication media are not generally suitable for ad hoc and sensor networks. Despite the higher degree of control and the higher signal quality for wired communications, this communication medium lacks the flexibility required by mobile and transient applications that characterize ad hoc and sensor networks. In many cases, the mere reliance on wired communications implies the need for installation and deployment of some form of infrastructure, violating the basic premise of ad hoc and sensor networks. Furthermore, the use of wires severely limits the mobility of a system by the length of wires. Finally, many ad hoc and sensor network applications require deployment in situations where wire installation is not practical, such as disaster relief or environmental monitoring.

The above discussion has shown that wired communications could be useful for particular ad hoc and sensor applications, but it is not suitable for the general application space of these networks. The next section focuses on wireless communications, which overcome many of the drawbacks of wired communications for ad hoc and sensor networks.

2.1.2 Wireless Communication

Wireless communications rely on signal transmission over a medium without the presence of wires or cables between the sender and receiver. Possible communication media for wireless communication include air, water, or vacuum. Wireless communications can support a high degree of mobility and deployment flexibility, so they are the main communication medium of choice for ad hoc and sensor networks.

The attractive feature of wireless communications, the absence of wires, also presents drawbacks. The absence of a physical wire connecting the sender and receiver render the transmitted signal much more vulnerable to interferences and background noise while traversing the wireless medium. As a result, the expected signal quality of a wireless communication link is relatively lower, less stable, and less predictable than a comparable wired link. The higher vulnerability to interferences requires higher quality margins and smarter control of wireless links to maintain communication. Wireless communications are also inherently less secure than wired communications. An eavesdropper simply needs to capture the wireless signal through an available receiver, whereas listening in to wired communications requires physically tapping into the communication line. The use of wireless communications also complicates higher layer network functionality, such as the hidden terminal problem at the MAC layer, which is discussed further in Ch. 3. Wireless communications require more sophisticated and adaptive mechanisms at several layers of the network stack.

The flexibility, practicality, and support for mobility of wireless communications overweigh the drawbacks discussed above. The wide scope of potential applications for wireless communications, especially in the context of ad hoc and sensor networks, warrants the added development and operating cost for advanced network management mechanisms.

This section has covered the potential communication media for ad hoc and sensor networks. The next section explores the communication technologies that utilize the medium.

2.2 Communication Technologies

A system's communication technology specifies the energy type for encoding information bits, as well as the methods for encoding information bits into energy and decoding them. Examples of communication technologies include radio frequency, infrared, microwave, laser, ultra wide band radio, and acoustics. The adoption of different communication technologies stems from the diverse needs of communication applications. For example, microwave and infrared technologies provide point-to-point links between a sender and receiver, yielding better communication efficiency and signal quality. However, ad hoc and sensor network applications may benefit more from broadcast communication technologies, such as radio frequency and ultra wide band radio. In this section, we survey the potential communication technologies and their suitability for ad hoc and sensor networks. Section 2.2.1 discusses the technologies that typically use point-to-point communication, while Section 2.2.2 focuses on broadcast communication technologies.

2.2.1 Point-to-Point Communication Technologies

Point-to-point communication technologies have their roots in many wired communication applications, such as telephone networks or long-distance data transmission lines. The main purpose of point-to-point communication technologies is to establish a one-to-one communication link between a sender and the intended receiver. Achieving this property through wires is relatively simple, since wires can physically guide the signal along its designated path.

Point-to-point communication through wireless technologies is a more challenging task. The wireless medium is inherently a broadcast medium in which the signal of a wireless transmitter spreads in all outbound directions. Due to the inherent broadcast nature of many wireless communication technologies, directional antennas are used to guide the transmitter's signal energy towards the receiver. Directional antennas provide a higher signal quality at the receiver by channeling most of the energy in the direction of the receiver. The drawback of directional antennas is the higher cost and hardware complexity. Even with the use of directional antennas, most wireless point-to-point communication technologies also require an unobstructed line-of-sight (LOS) between the sender and receiver.

Network applications have used certain communication technologies, such as infrared and microwave signals, for point-to-point wireless communications. Infrared technology encodes information through signals with a wavelength between 750nm and 1mm, the so-called infrared spectrum. For example, many laptops are equipped with built-in infrared ports for interfacing with cell phones and other laptops. When a laptop comes into the vicinity of another device equipped with infrared communication capability, the two devices can establish a communication link. The infrared port of the two devices must be closely aligned without any physical obstacles between them to ensure a LOS. Another common example of one-way infrared wireless communications is television remote controls. The disadvantage of using infrared technology for ad hoc and sensor networks, in addition to the LOS requirement, is its susceptibility to interference from light sources, such as neon lights or sunlight. Furthermore, available infrared transceivers have limited communication ranges within the order of tens of meters, which constrains network range.

Microwave technology uses high frequency radio signals, with wavelengths ranging between 1mm to 30cm. Microwave technology can be either a point-to-point or a broadcast technology. A common application of point-to-point microwave technology is the provision of television signals to subscribers through small dishes for signal transmission and reception. Point-to-point microwave applications are highly directive, requiring a careful alignment and maintenance of orientation between the sender and receiver. In urban areas, microwave transceivers are generally installed on roofs in order to ensure a LOS from an antenna tower, because an obstruction in the LOS severely affects the communication.

In general, point-to-point wireless communication technologies require precise positioning and orientation of the transceiver to maintain acceptable communication links. This property renders point-to-point technologies suitable for a small and specific subset of ad hoc and sensor network applications, namely scenarios with limited mobility and highly predictable topologies. The next section discusses the class of communication technologies that is more suitable for ad hoc and sensor networks: broadcast technologies.

2.2.2 Broadcast Communication Technologies

Broadcast communication technologies support the concurrent reception of a transmitted signal by multiple receivers. In contrast to point-to-point technologies that require careful positioning and alignment of the transceivers, broadcast communications can use lower complexity omnidirectional antennas that require much less maintenance, so they provide better support for ad hoc deployments and mobile networks. The above properties of broadcast technologies have made them the top choice for ad hoc and sensor networks.

A related technology is satellite communications through which a network of artificial satellites orbiting the earth relays earth-based signals. Satellite communications currently support telephone, television, radio, scientific, and military applications. Satellites inherently represent infrastructure networks, since satellite deployment involves extensive planning and high deployment cost for putting the nodes into orbit. In the context of ad hoc and sensor networks, satellites can serve as the supporting infrastructure for the network.

For example, many ad hoc and sensor network design approaches consider that each network node possesses location information through the satellite-based Global Positioning System (GPS) [4].

One of the more established broadcast communication technologies is through radio frequency (RF) waves. Sending radio frequency waves entails feeding alternative current to an antenna to produce electromagnetic waves. The RF spectrum includes frequencies from a few hertz to several hundred gigahertz. Applications that use RF technology include radio, television, cellular telephones, and radar. A large portion of the current standards that are applicable for wireless networks, and especially ad hoc and sensor networks, uses RF waves. The popular IEEE 802.11 standard [5], which supports both centralized and ad hoc modes for wireless local area networks, relies on RF waves in both the 2.4 Ghz and the 5 Ghz bands. Similarly, the recent Bluetooth standard [6] for wireless personal area networks (WPAN) also uses RF waves in the 2.4 Ghz band.

Many sensor network manufacturers have also adopted RF communication technology. For example, the widely used Crossbow mica motes [7] have adopted RF communication in the 400Mhz, 900Mhz, and 2.4Ghz bands. The latter band satisfies the recent Zigbee [8] standard for sensor networks. Radio frequency identification (RFID) [9], an emerging technology for replacing bar codes through tiny radio frequency tags, represents another application for RF communication. Chapter 10 covers a case study of an RF sensor network.

Another emerging RF technology is ultra wide band (UWB) radio, a spread-spectrum technique based on the modulation of short nanosecond low power pulses [10]. This technology has been used for radar applications for over half a century. In recent years, UWB has received increasing recognition for its applicability to short range communication networks because of desirable features such as high data rates, low power consumption, precise ranging capability, resistance to multipath fading, and penetration of dense objects. All of the above properties make UWB a strong candidate technology for ad hoc and sensor networks. For example, emergency workers using an UWB ad hoc network for earthquake recovery could place nodes equipped with sensors in the rubble to detect signs of living survivors. Because of UWB's ground penetrating capability, the nodes in the rubble can effectively communicate with surface nodes. Chapter 11 presents an example scenario of UWB ad hoc networks.

Acoustic communication is yet another broadcast technology that has recently received increasing attention. Acoustic communication relies on the modulation of acoustic waves with digital data. While acoustics has been the technology of choice for underwater communications for over half a century [11], several projects have demonstrated the usefulness and applicability of acoustics for affordable and easily deployable mobile applications within the area of ubiquitous computing [13–16]. For instance, many mobile devices can exploit on-board speakers and microphones to communicate acoustically. Acoustic waves typically have a short communication range, and they do not penetrate walls, which adds security to the communication. On the downside, the supportable information transfer rate of acoustics is limited by the narrow acoustic bandwidth. Chapter 12 provides an example of how ad hoc and sensor networks can exploit the low bit rate and short communication range capabilities of acoustic communications to form multihop networks.

Broadcast communication has commonly been associated with wireless media, but there are also networks that employ broadcast communications over wired media. For example, Ethernet networks that work over category 5 cables broadcast signals over wires and through hubs. The broadcast nature of Ethernet necessitates mechanisms at higher layers to avoid collisions.

2.3 Physical Layer Optimization Parameters

This section identifies the relevant parameters at the physical layer, which can be incorporated into cross-layer design strategies.

2.3.1 Transmission Power

In wireless communications, the transmitter emits signals at a certain power level, which is referred to as the transmission power. The signal loses energy as it propagates from sender to receiver. The so-called signal path loss varies proportionally with d^α, where d is the distance between sender and receiver, and α is the path loss coefficient ranging between 2 and 4. The transmission power must be high enough to achieve an acceptable signal quality at the receiver. However, the transmission power is also upper bounded by regulatory limits and by interference considerations at neighboring transceivers. Transmitters must use a power level within these constraints i.e. a power level that is both sufficient to communicate effectively with the receiver and that adheres to regulatory emission limits. Because the medium conditions in ad hoc and sensor networks are highly dynamic, nodes should ideally adapt their transmission power continuously to the current conditions. Cross-layer design can enable interaction between the physical layer and higher layers for better transmission power adaptation.

2.3.2 Processing Power

Many of the traditional network protocols do not consider processing power in determining network behavior. However, processing power can play a significant role for ad hoc and sensor network protocols. Most ad hoc networks employ a multihop communication strategy with a short distance per link. For networks with short range wireless links, the processing power becomes non-negligible relative to the transmission power. While it is difficult to enforce strict processing power control at run-time, consideration of processing power in determining network behavior can improve performance. For example, some sensor networks support in-network processing. A load balancing strategy for these sensor networks must consider processing power at each node to determine how to distribute the network load evenly. Because load balancing typically occurs at higher layers, a cross-layer design strategy is required to expose the processing power information to higher layers.

2.3.3 Sensing Power

In sensor networks, the sampling of physical indicators also consumes power, referred to as the sensing power. As for the case of processing, sensing power becomes appreciable relative to transmission power for shorter wireless links. For instance, consider a seismic monitoring sensor network in which nodes periodically sample their sensors to determine if the seismic activity is above a certain threshold level. If so, then the nodes communicate the sensed data towards the user. Otherwise, the node continues the periodic sampling of their sensors until an event occurs. If a long time passes before the occurrence of a seismic event, the network nodes do not consume power due to transmissions during that time. However, the nodes do consume sensing power for periodic sampling of the sensors.

2.3.4 Signal-to-Noise Ratio

The signal-to-noise ratio (SNR) is a quality indicator for communication links. The SNR provides a figure of merit through the comparison of the received signal strength with noise level at the receiver. Thus, the SNR is proportional to the received power and inversely proportional to the sum of the background noise and interference at the receiver. Stated differently, improving the signal quality at a receiver can be achieved either by increasing the transmission power (which causes an increase in the received power) or by reducing background noise and interference. Typically, wireless applications set minimum requirements for SNR on a network-wide or a per-link basis. In cross-layer design, the individual link SNR can serve as an input to a comprehensive optimization of node behavior that satisfies the physical layer quality requirements.

2.3.5 Transmission Rate

The transmission rate indicates the current transfer rate of a communication link. Transmission rates are closely related to the transmission power. Consider an active communication link that satisfies the SNR quality requirements. Increasing the link's rate while maintaining the SNR unchanged requires an increase in the transmission power. Network mechanisms can exploit this relationship to trade off lower rates for a reduced transmission power for rate-elastic traffic. Similarly, nodes can achieve higher transmission rates through an increase in transmission power.

2.3.6 Modulation Code and Rate

Modulation is the process of varying a carrier signal in order to use that signal to convey information. Three basic features of the signal can be varied to carry information: amplitude, frequency, or phase. In addition, modulation techniques can use a combination of these features. For example, Pulse Position Modulation (PPM) is a common modulation technique for time-hopping UWB networks. UWB relies on the regular transmission of nanosecond signals, called monocycles. To encode information, PPM shifts monocycles in time. For example, sending the monocycle 1 ns earlier indicates a zero bit, and delaying the monocycle by 1 ns indicates a one bit. M-ary PPM can also encode several bits per monocycle, by defining 2^M shift positions of the monocycle. The number of bits encoded in each PPM symbol is referred to as the modulation rate. Increasing the modulation rate yields increases in the transmission rate, but it also lowers the signal quality since it makes it more difficult for the receiver to decode the signal.

Adaptive cross-layer mechanisms can vary the modulation rate according to dynamic medium conditions. For example, if a node observes a rise in the interference level, it can lower its modulation in order to ensure that the receiver can still decode the signal.

A common feature of spread-spectrum technologies, such as UWB or Code Division Multiple Access (CDMA) [17], is the use of codes to provide signal robustness and security. Spread-spectrum technologies enable concurrent transmission through the use of codes that are orthogonal or quasi-orthogonal. In a network with ongoing links, selecting an appropriate code for a new link can maximize the rate for the new link and minimize the impact on the perceived interference of neighboring nodes.

3

Data Link Layer

The second layer in the OSI reference model is the data link layer, shown in Fig. 3.1. The link layer handles access to the underlying channel and defines the data format. It is responsible for establishing the physical and logical connection between nodes. The link layer is further split into two sublayers: the Medium Access Control (MAC) layer, and the Logical Link Control (LLC). Most of the challenges of ad hoc and sensor networks, such as efficient and distributed control of the channel, occur at the MAC layer. This has motivated ad hoc and sensor network research to focus more on the MAC layer, which is the main focus of this chapter. At the end of the chapter, we provide a brief discussion on LLC issues relating to ad hoc and sensor networks.

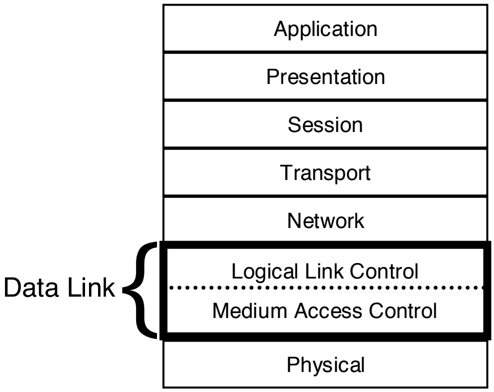


Fig. 3.1. The data link layer in the OSI reference model

Portions reprinted, with permission, from (R. Jurdak, C. V. Lopes, and P. Baldi. "A Survey, Classification, and Comparative Analysis of Medium Access Control Protocols for Ad Hoc Networks". IEEE Communications Surveys and Tutorials, 6:1) ©2004 IEEE.

3.1 Introduction

In the OSI reference model, medium access is a function of the layer 2 sublayer called the Medium Access Control (MAC) layer. MAC protocols for wireless networks must address the hidden node problem (discussed in Sect. 3.2.1) and must exercise power control. Accessing the wireless medium thus requires more elaborate mechanisms than wired networks to regulate user access to the channel. The absence of a centralized controller in wireless ad hoc and sensor networks presents even greater MAC layer challenges than infrastructure wireless networks, creating a need for distributed management protocols at the MAC layer, and possibly at higher layers of the network stack.

Here, we analyze the features of ad hoc and sensor network MAC protocols through a comprehensive survey of existing MAC protocols, ranging from industry standards (IEEE 802.11 [5], Zigbee [8], and Bluetooth [6]) to research proposals. Table 3.1 shows the list of surveyed MAC protocols for wireless ad hoc and sensor networks in chronological order.

Designing improved protocols at the MAC layer requires an understanding of the features that characterize such protocols. From our survey of available MAC protocols for ad hoc and sensor networks, 5 key features emerge: (1) channel separation and access; (2) transmission initiation; (3) topology; (4) power; and (5) traffic load and scalability. A detailed justification on the selection of these features is available in [179].

3.1.1 Protocol Overview

It is evident from the overview presented above that a variety of design choices can be made for each feature and application. Combining various design choices of features involves complex tradeoffs. In addition, most protocols in Table 3.1 were designed for a specific class of applications or physical layer technologies, thus trading off generality for efficiency. Here, we analyze these tradeoffs for the existing protocols, and we further assess the suitability of various combinations of features for ad hoc network and sensor network applications. This tradeoff analysis and the classification of the protocols yield appropriate design guidelines for general wireless ad hoc and sensor network MAC protocols.

In the rest of the chapter, each section focuses on one of the protocol features through a discussion of representative protocols. Note that we drop the term "wireless" when referring to ad hoc and sensor networks in these sections. Section 3.2 describes existing channel separation and access techniques, which are the central mechanisms of MAC protocols. Section 3.3 focuses on the transmission initiation feature and discusses the effect that this feature has on a protocol's performance and applications.

The subsequent sections discuss the additional features that exploit cross-layer information for improving the performance of ad hoc network and sensor

Protocol	Channel	Topology	Trans. Initiation	Power Efficient	Traf. Load and Scal.
1. CSMA [24]	Single	Single/Flat	sender	no	Wired Networks
2. BTMA [25]	1 Control/1 Data	Centralized	sender	no	Hidden Terminal
3. PRMA [27]	Hybrid	Centralized	sender	no	Voice
4. MACA [28]	Single	Single/Flat	sender	no	Hidden Terminal
5. MACAW [29]	Single	Centralized	sender	no	Delivery Guarantee
6. FAMA [30]	Single	Single/Flat	sender	no	Delivery Guarantee
7. IEEE 802.11 [5]	Mult.(CDMA)	Single/Flat	sender	no	Access Point
8. HIPERLAN [18]	Mult.(Hybrid)	Clustered	sender	yes	Data Relay
9. MACA-BI [31]	Single	Multiple/Flat	receiver	no	Predictable Traffic
10. FPRP [32]	Multiple(TDMA)	Multiple/Flat	sender	no	Voice
11. PAMAS [96]	1 Control/1 Data	Multiple/Flat	sender	yes	Dense Low Load
12. Bluetooth [6]	Multiple (CDMA)	Clustered	master	yes	Low Rate PAN
13. Markowski [34]	Multiple(TDMA)	Single/Flat	N/A	yes	Voice
14. HRMA [35]	Hybrid	Multiple/Flat	sender	no	Large Packets
15. MCSMA [36]	Multiple(FDMA)	Single/Flat	sender	no	High Density
16. PS-DCC [37]	Single	Single/Flat	sender	yes	High Load
17. RIMA-SP [38]	Single	Single/Flat	receiver	no	Predictable Traffic
18. ADAPT [39]	Multiple(TDMA)	Multiple/Flat	sender	no	High Load
19. CATA [40]	Multiple(TDMA)	Multiple/Flat	sender	no	Low Load
20. Jin [41]	Hybrid	Clustered	sender	yes	Heterogenous
21. MARCH [42]	Single	Multiple/Flat	sender	Implicit	Homogeneous
22. RICH-DP [43]	Multiple(CDMA)	Multiple/Flat	receiver	no	High Load
23. SRMA/PA [44]	Multiple(TDMA)	Multiple/Flat	sender	yes	Voice
24. DCA-PC [45]	1 Control/N Data	Multiple/Flat	sender	yes	High Density
25. GPC [46]	Single	Clustered	N/A	yes	High Density
26. VBS [47]	N/A	Clustered	N/A	no	Voice
27. DPC/ALP [48]	Single	Multiple/Flat	sender	yes	Heterogenous
28. Lal [49]	Multiple(SDMA)	Multiple/Flat	receiver	Implicit	High Load/Density
29. GRID-B [50]	1 Control/N Data	Multiple/Flat	sender	no	High Load/Density
30. MC MAC [51]	Multiple(CDMA)	Multiple/Flat	sender	no	High Rate PAN
31. WCA [52]	N/A	Clustered	N/A	yes	Heterogeneous
32. DBTMA [53]	2 control/1 data	Multiple/Flat	sender	no	Hidden Terminal
33. MMAC [54]	Multiple(SDMA)	Multiple/Flat	sender	yes	High Load
34. D-PRMA [55]	Multiple(TDMA)	Single/Flat	sender	no	Voice
35. SMAC [57]	Multiple (TDMA)	Single/Flat	sender	yes	Long deployment
36. T-MAC [58]	Multiple (TDMA)	Single/Flat	sender	yes	Long deployment
37. BMAC [56]	Single	Single/Flat	sender	yes	Long deployment

Table 3.1. Protocol Classification

MAC protocols. Section 3.4 examines the effect of incorporating topology information on the MAC protocol performance. Section 3.5 assesses available power management mechanisms and their suitability for particular channel access methods and topologies. Section 3.6 evaluates the scalability and performance of MAC protocol design choices. Section 3.8 offers a roundup of ad hoc and sensor network MAC issues and derives guidelines for creating a more generalized protocol that is suitable for several physical-layer technologies and applications.

3.2 Channel Separation and Access

A key factor in the design of a MAC protocol for ad hoc and sensor networks is the way in which it utilizes the available medium. Earlier approaches assumed a common channel for all stations, while more recent approaches have used multiple channels for more efficient use of the medium.

In this section, we explore both single channel and multiple channel MAC protocols. Furthermore, we classify multiple channel protocols based on their channel separation mechanism. Within each channel separation strategy, we describe the channel access method of particular protocols.

3.2.1 Single Channel

Considering the medium as a single channel was the most prominent approach in the earlier years of MAC design [24, 26, 28–30], primarily because mechanisms for channel separation had not yet been developed. In a common channel MAC protocol, all the nodes in the network share the medium for all their control and data transmissions. Collisions are an inherent attribute of such protocols. Two stations that transmit simultaneously will both fail, and a back-off mechanism is required by both stations.

The first proposed single channel protocol is Carrier Sense Multiple Access (CSMA) [24]. In CSMA, a node senses the common channel for ongoing transmissions. If the channel is idle, it begins its transmission. Otherwise, it sets a random timer before attempting to transmit again. CSMA does not address the handling of collisions on the channel. An improved variant of CSMA is CSMA/CD [26] (CSMA with collision detection). In CSMA/CD, if two or more transmissions collide, the sending nodes are notified and each chooses a random time before retransmitting. If a node detects a collision for the second time, it backs off for twice the time it backed off the last time. This mechanism is known as Binary Exponential Back-off (BEB). The performance of CSMA protocols degrades quickly with high load, due to increased frequency of collisions and increased transmission latency.

When applying CSMA to networks where some nodes are not within range of each other, two or more nodes may have a common neighbor while they are out of range. If both nodes sense the channel and try to transmit to this common neighbor, then a collision occurs. Figure 3.2 illustrates this situation, which is called the hidden node problem. Multiple Access with Collision Avoidance (MACA) [28] was proposed for packet radio networks as an improvement of CSMA to eliminate the hidden terminal problem. The protocol introduces a handshake between a sender and receiver, shown in Fig. 3.3. This handshake ensures that neighboring nodes are aware of the upcoming transmission, and that they will refrain from sending for the duration of that transmission. The sender initiates the handshake by transmitting a Request to Send (RTS) signal to the receiver to indicate its request to access the medium. Nodes in the vicinity of the sender are notified of the upcoming transmission through this RTS message. Upon receiving a RTS, the receiver replies with a Clear to Send (CTS), indicating its readiness for reception. Nodes that are in the vicinity of the receiver are also notified of the transmission through the CTS. Once the RTS/CTS handshake is complete, the transmission proceeds with no risk of collisions. If there is a collision of two RTS messages, then both stations back off for some time. By reducing the possibility of collisions and eliminating the hidden terminal problem for data transmissions, MACA offers an improvement over CSMA. MACA Wireless (MACAW) [29] was introduced to adapt MACA for the unreliability of the wireless medium, by making the receiver acknowledge successful data reception with an ACK message. This offers a delivery guarantee that is crucial in wireless networks. MACAW is based on a cellular structure in which a base station resides in each cell, and base stations are interconnected by a wired network. An additional modification in MACAW is replacing the BEB mechanism with a smaller back-off factor[1]. This modification aims at reducing the latency caused by frequent collisions in loaded networks. Floor Acquisition Multiple Access (FAMA) [30]

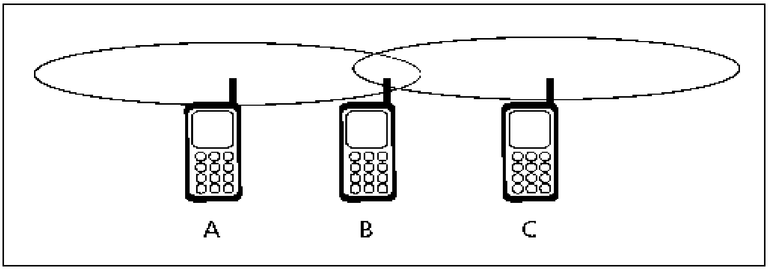


Fig. 3.2. The Hidden Node Problem: Node A senses the medium as idle and initiates a transmission to node B. Node C also senses the medium as idle and initiates a transmission to node B. A collision occurs at node B, and both A and C are unaware of the collision since they are out of each other's range

---

[1] BEB was reduced to use a factor of 1.5 because of the high latency exhibited for a factor of 2. The optimal back-off factor could be obtained adaptively based on current channel utilization. [37]

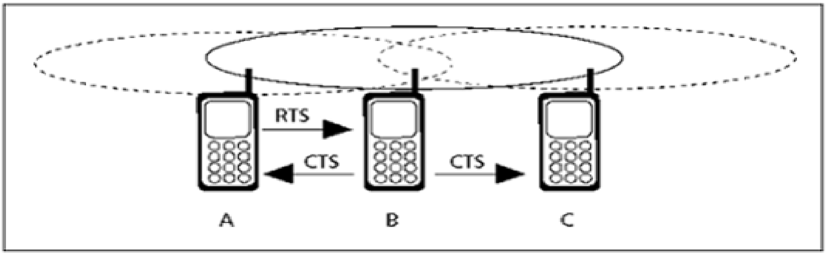


Fig. 3.3. RTS/CTS handshake: Node A requests access of the channel through the RTS. Node B replies with a CTS indicating that it is ready to receive node A's transmission. Node C receives a CTS from node B and thus refrains from transmitting for the duration indicated in CTS. Even though A and C are hidden from each other, the handshake ensures that a collision at node B does not occur

enhances MACAW by adding carrier sensing before sending a RTS. MACA-BI (By invitation) [31] takes a receiver-initiated approach, where a receiver indicates its readiness to receive by broadcasting a Ready to Receive (RTR) message. Any neighbor that hears a RTR can then send data to any destination. Therefore, MACA-BI does not prevent collisions in the vicinity of the receiver.

Receiver Initiated Multiple Access with Simple Polling (RIMA-SP) [38] improves on MACA-BI by allowing polled neighbors to send only to the polling node. RIMA-SP also allows both nodes to send data after the handshake is complete. In both MACA-BI and RIMA-SP, the receiver takes a proactive role in initiating transmissions. Transmission initiation will be discussed further in section 3.3.2.

Multiple Access with Reduced Handshake (MARCH) [42] attempts to reduce control signaling, while retaining the RTS and CTS framework. The handshaking involved in MARCH is shown in Fig. 3.4. Suppose node A has data to send to node Z, using a path A,B,C,D,Z. A sends RTS_A to the next hop in the path B. When B replies to A with CTS_B, C hears that message. C now knows that B will send it data from A, so it will reply with CTS_C to B at the appropriate time. The same process is repeated at nodes D and Z. Using this mechanism, MARCH proposes a single RTS on the first hop of the path, while only CTS is required for every subsequent hop.

Distributed Power Control with Active Link Protection/Adaptive Probing (DPC/ALP) [48] also relies on the basic RTS/CTS handshake. In DPC/ALP, the sender issues a RTS at a power level that appears as noise, and keeps progressively increasing power and sending it again until the receiver replies with a CTS. If the transmit power for a RTS exceeds a threshold with no reply from the receiver, the sender backs off. This mechanism allows a RTS to interfere only minimally with other ongoing transmissions, since the signal will barely exceed the noise power.

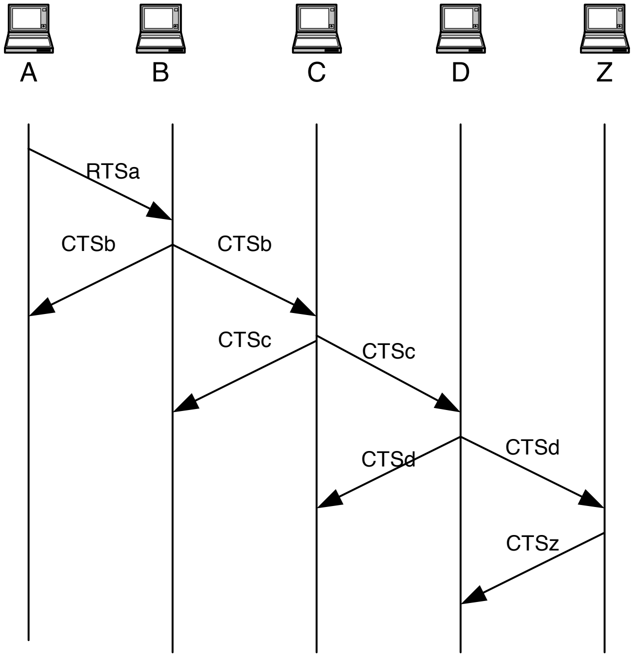


Fig. 3.4. Reduced handshaking in MARCH

Power Save with Distributed Contention Control (PS-DCC) [37] is designed as a probabilistic back-off mechanism for each channel in IEEE 802.11 [5] networks. In PS-DCC, nodes measure channel utilization constantly and adaptively calculate a sending probability based on the current network load. PS-DCC offers better performance than a static back-off scheme, such as the one used in most deployed protocols.

More recently, a prominent single channel MAC protocol targeting sensor networks has emerged. Because sensor nodes are extremely resource-limited, single channel MAC protocols are attractive for sensor networks due to their simplicity. A single channel MAC avoids using complex mechanisms or hardware. BMAC [56] is a modular and flexible sensor network MAC protocol which aims at reducing idle listening, which is a major cause of power consumption in long-term monitoring sensor networks. BMAC enables each node to wake up periodically to check for channel activity. The wake-up period is referred to as the check interval. BMAC defines 8 check intervals, and each check interval corresponds to one of BMAC's 8 listening modes. To ensure that all packets are heard by the nodes, packets are sent with a preamble whose reception time is longer than the check interval. BMAC therefore defines 8 different preamble lengths referred to as transmit modes.

3.2.2 Multiple Channels

Some protocols for ad hoc and sensor networks separate the control and data planes by assigning one channel for control signaling, and one or more separate channels for data transmissions. In this section, we focus on multiple channel protocols and we classify protocols according to their channel separation techniques.

Generalized Separation

Some multiple channel protocols describe a generalized channel separation scheme. Busy Tone Multiple Access (BTMA) suggests having a separate busy tone channel to solve the hidden terminal problem of CSMA, where a centralized base station sensing the data channel as busy can place a sine wave on the busy tone channel to prevent any nodes from transmitting. A recent extension of using busy tones was presented in Dual Tone Busy Tone Multiple Access (DT-BTMA), where two out-of-band busy tone channels are used to protect RTS transmission, and to prevent nodes in the receiver's vicinity from transmitting. It therefore focuses on solving the hidden terminal problem in a similar way to BTMA, while using a distributed approach rather than a base station. Power Aware Multiple Access with Signaling (PAMAS) proposes using one control channel for sending RTS/CTS, and a separate data channel. In terms of handshaking, PAMAS uses the same sequence as MACA. PAMAS also specifies that nodes that detect RTS or CTS refrain from communicating for the duration indicated in the overheard control messages.

Dynamic Channel Assignment with Power Control (DCA-PC) is another generalized channel separation protocol having one control and N data channels. In DCA-PC, a sender checks if any of the data channels appear free. If so, it chooses one of the available channels and sends a RTS signal on the common control channel with maximum power to the destination. If the destination agrees on the sender's channel choice with no conflict, it replies with CTS at a power level appropriate to reach the sender, and then the sender can reserve the channel. If the destination has a conflict with the sender's channel choice, the destination's free channel list is sent to the sender so that it can choose a more appropriate channel.

Another generalized channel separation protocol is Grid with Channel Borrowing (GRID-B), which proposes initially assigning channels to each cell in a predefined geographic area. Highly loaded cells would borrow channels from neighboring lightly loaded cells if needed. Negotiations for such lending would occur on a common control channel. GRID-B proposes the use of Code Division Multiple Access (CDMA) or Frequency Division Multiple Access (FDMA) for channel allocation. In the case of CDMA, channel bandwidths are fixed and therefore increasing the number of channels up to a certain limit is quite beneficial. In FDMA, the total bandwidth is fixed, and therefore having additional users would reduce the per user bandwidth.

Time Division Multiple Access

Time Division Multiple Access (TDMA) segments the medium into several fixed time frames that are subdivided into slots. To ensure that nodes keep track of time frames and slots, TDMA protocols must maintain synchronization among the nodes. In these protocols, only one station may transmit during a particular time slot. Because of their periodic nature, TDMA protocols are most suitable for real-time and deadline sensitive traffic.

The first proposed TDMA protocol for ad hoc networks is the Five Phase Reservation Protocol (FPRP) [32], in which each slot is split into an information slot and reservation slot. A sender that wants to reserve an information slot must contend for it during its reservation slot. The reservation slot consists of five phases that resolve conflicts among all nodes that are also contending for the information slot within a two-hop radius. A node that reserves an information slot can transmit with a low chance of collisions during that slot. In FPRP, nodes maintain perfect synchronization through GPS.

Collision Avoidance Time Allocation (CATA) [40] adopts almost the same concept as FPRP, with the only distinction of using four reservation mini-slots to access a slot instead of five. Soft Reservation Multiple Access with Priority Assignment (SRMA/PA) [44] also resembles FPRP, since they both share the notion of having several mini-slots to reserve a data slot. The added feature in SRMA/PA is that it classifies nodes into high and low-priority nodes, where high-priority nodes can grab reserved slots from low-priority nodes. Categorizing nodes in this manner gives better performance for voice nodes in the network. The protocol also suggests a new back-off mechanism, where access probability is based on packet laxity.

Markowski [34] proposed a window splitting protocol based on TDMA. This protocol classifies nodes according to their traffic classes: Hard Real Time (HRT), Soft Real Time (SRT), and Non Real Time (NRT). Each class of nodes can preempt nodes in lower classes. Furthermore, nodes are only allowed to transmit at the beginning of a slot, while all nodes maintain perfect synchronization. Within each class, collisions are resolved through a window splitting mechanism. If a collision occurs for two nodes of the same class, then half of the nodes of that class are placed in an active window for the current slot, while the other half are placed in an inactive window. Nodes in the active window contend for the next slot. If collisions occur again, the active window is further split into an active and an inactive window. Window splitting is done on Node ID basis for HRT, on packet laxity for SRT, and on arrival time for NRT. There is no specification of how to handle synchronization in this protocol, nor is there any reference to the hidden node problem.

ADAPT [39] proposes assigning slots to nodes to cope with high load and high density networks. It also suggests using contention to manage the unused slots. Each active node owns one slot and is given priority to send RTS in its slot while other nodes listen for the owner's transmission. If the owner does not send a RTS during its own slot, other nodes will contend for this slot by trying to send their own RTS. At that point, a node that receives a CTS message may use this slot only in the current frame.

In Distributed Packet Reservation Multiple Access (D-PRMA) [55], which is an adaptation of PRMA (see Section 3.2.2), nodes are designated as voice or data terminals. Only voice terminals can reserve the same slot for subsequent frames. This resembles slot ownership in ADAPT with the distinction that only voice terminals may temporarily own slots. A slot is split into m mini-slots, and contention in the first mini-slot determines the winner in the slot. If there are collisions in the first mini-slot, nodes contend again for the slot during the second mini-slot, and so on. Voice terminals are given priority over data terminals by always contending through the first mini-slots. Synchronization in D-PRMA is achieved through GPS.

There are also TDMA-based MAC protocol that are specifically designed for sensor networks. Sensor MAC (S-MAC) [57] is a heavyweight MAC protocol for sensor networks that relies on time synchronization and scheduling among nodes to enforce periodic sleep and listen schedules. S-MAC uses three novel techniques to reduce energy consumption and support self-configuration:

1. To reduce energy consumption in listening to an idle channel, nodes periodically sleep. Neighboring nodes form virtual clusters to auto-synchronize on sleep schedules.
2. S-MAC also sets the radio to sleep during transmissions of other nodes. Unlike PAMAS, it only uses in-channel signaling.
3. S-MAC applies message passing to reduce contention latency for sensor network applications that require store-and-forward processing as data move through the network.

A more recent version of S-MAC [190] has introduced adaptive duty cycles by enabling a node to snoop on neighbors' RTS and CTS messages in order to schedule its own wake-up time. Because nodes have to maintain neighbors' schedules, S-MAC is complex and not sufficiently scalable for large scale networks of resource-limited nodes.

T-MAC [58] also proposes adaptive duty cycles to address the nonuniform traffic patterns in sensor networks. T-MAC is similar to S-MAC in its essence, but it introduces early sleeping to enable nodes that are scheduled to be active to go into sleep mode if they are idle. T-MAC suffers from similar complexity and scaling problems of S-MAC, because it trades off a short active time for reduced adaptivity to changing network conditions.

Frequency Division Multiple Access

FDMA splits the available medium into several frequency channels to allow multiple nodes to transmit simultaneously. A proposed FDMA protocol uses CSMA on each of the frequency channels (MCSMA) [36]. Each node keeps a list of free channels, and when it has data to transmit, it tries to use the channel that it used during the last transmission. If that channel is busy, it selects one of the other free channels. Although MCSMA reduces overall collisions when compared to original CSMA, collisions and hidden terminal problems are still present on each channel.

Code Division Multiple Access

CDMA uses one of several orthogonal codes to spread each sender's signal. Through its use of orthogonal codes, CDMA allows concurrent multiple transmissions using all of the available spectrum. Multi-code MAC (MC MAC) [51] uses CDMA by assigning N codes for data transmission and one common code for control signaling. A sending node in MC MAC issues a RTS to another node on the common control channel indicating the code(s) that it will use for transmission. If it gets a CTS it assumes that there is no code conflict with the intended receiver, and it subsequently sends data after which it expects an ACK. If the receiver detects code conflicts, it exchanges its usable codes with the sender, so the sender chooses the appropriate codes for transmission.

IEEE 802.11 [5] Distributed Coordination Function (which is the specification for infrastructureless mode in Wireless Ethernet) is almost identical to MACAW on each of its channels, except that it combines a CSMA mechanism with MACAW to lower the probability of RTS collisions. IEEE 802.11 splits the medium by using one of two forms of CDMA, either Frequency Hopping Spread Spectrum (FHSS), or Direct Sequence Spread Spectrum (DSSS). In FHSS, it allows up to 79 different hopping channels in North America and Europe, thus supporting up to 26 co-located networks. In DSSS, IEEE 802.11b uses 12 codes to allow concurrent transmissions of nodes. In IEEE 802.11, nodes that hear RTS or CTS set their Network Allocation Vector (NAV), which indicates the remaining time until a current channel becomes free. The standard also provides a busy channel which nodes can sense to check if the medium is idle. To promote fairness among nodes and to prevent large transmission latencies, IEEE 802.11 introduces a contention window from which nodes waiting to transmit choose a random back-off time. The size of the contention window adapts to the number of collisions that occur. A node may transmit once its back-off timer expires. Whenever a node is forced to wait through another frame, it continues counting down from where it stopped instead of choosing a new random waiting time. This ensures that nodes that wait longer get priority to access the medium.

Receiver Initiated Channel Hopping with Dual Polling (RICH-DP) [43] combines the slow frequency hopping aspect of HRMA [35] and the receiver initiation aspect of RIMA/SP [38] to allow nodes to reserve hops and to send data both ways once a hop is reserved. A receiver that is ready to receive data sends a RTR message to its neighbors. If some neighbor has data to send, it responds with a RTS to reserve the hop for data exchange between the pair. Both nodes can send data once they complete this reservation.

Space Division Multiple Access

Like in CDMA, Space Division Multiple Access (SDMA) aims at using the full spectrum all of the time to a certain degree. In SDMA, nodes use directional antennas thus allowing a node to begin a transmission at any time as long as that the transmission's direction does not interfere with an ongoing transmission. In Lal's [49] SDMA protocol, a node polls neighbors with an omnidirectional RTR message that contains the node's training sequence. A training sequence indicates the directions in which a node accepts transmissions. Nodes that have data to send reply to the polling node using a directional RTS (DRTS) that contains each node's own training sequence. The polling node then replies to the accepted senders with a Directional CTS (DCTS) to complete the handshake. The use of directional antennas can therefore accommodate concurrent multiple senders to the polling node.

MMAC [54] proposes a technique of using smart antennas to establish multihop link through the RTS/CTS handshake. In MMAC, nodes keep profiles of the neighboring transceiver directions. Whenever a new data request arrives at the MAC layer, the MAC initiates carrier sensing in the direction of the intended receiver. If the channel is idle in that direction, the MAC then issues a directional RTS to the next hop in the path to the destination. Nodes on the path to the destination forward this RTS message directionally until the RTS reaches its destination. The destination then replies with CTS directly to the sender, and the two establish data communication. Neighboring nodes that fall within the range of this new directed link set their Directional Network Allocation Vector (DNAV) for the duration of the transmission.

Hybrid Protocols

Hybrid protocols combine two or more of the above approaches. Packet Reservation Multiple Access (PRMA) [27], which was designed for an infrastructure network to enable voice nodes to communicate alongside data nodes, uses both TDMA and FDMA. PRMA divides time into frames that are further segmented into slots, and each slot may be either reserved or unreserved. There is also one upstream and one downstream frequency channel. A sender has to listen for an unreserved slot, to contend for it using ALOHA, and to await a base station's decision on the winner for this slot. PRMA also classifies nodes as periodic and non-periodic traffic nodes. If a periodic traffic node reserves a slot, that node can use the same slot in subsequent frames.

High Performance Local Area Network (HIPERLAN) [18], which is the European counterpart of IEEE 802.11, uses another hybrid FDMA/TDMA channel access scheme. It provides a maximum of five frequency channels, each with a rate of 23.5 Mbps. Furthermore, nodes have to contend for the channel in three phases prior to reserving it. The length and structure of these phases are based on fixed time slots and frames.

Hop Reservation Multiple Access (HRMA) [35] is another hybrid of TDMA and FDMA. In HRMA, nodes are all synchronized and hop to a common sequence (each slot, all the nodes listen to the same frequency, and switch to another frequency during the next slot). If a node has data to send, it sends a Hop Reservation (HR) message using the current frequency hop, and it follows that with a RTS message to the intended receiver. If the receiver replies with CTS, then the pair has reserved the current frequency hop and they can send any amount of data using that frequency, while all other nodes are still following the common hopping sequence.

Jin [41] proposes a hybrid CDMA/TDMA protocol, where nodes dynamically elect a Pseudo Base Station (PBS) based on power considerations. The PBS maintains synchronization and assigns codes to nodes that it manages, using frames that have 3 mini-slots for synchronization, reservation, scheduling, in addition to the data slot.

Bluetooth [6] also combines CDMA and TDMA. A master node in Bluetooth assigns frequency hopping sequences a piconet, which allows simultaneous communication between the master and up to seven slaves. Bluetooth uses time division duplexity to separate the uplink and downlink. The master node manages medium access through a polling and reservation scheme, and it also assigns hopping sequences and maintains synchronization within the piconet.

3.2.3 Channel Separation and Access Summary

Many medium access protocols for ad hoc networks use a variant of the RTS/CTS handshake. Protocols that do not use this handshake rely on carrier sensing, periodic exchanges of information among nodes, or reservations. Multiple channel protocols, which use different techniques for channel separation, generally allow for more users than single channel protocols.

3.3 Transmission Initiation

Some protocols adopt a sender-initiated approach to transmissions; others select a receiver-initiated approach. The choice of a transmission initiation strategy depends on the protocol's target application types. Historically, sender-initiated protocols were most common until recently. In the rest of this section, we discuss each of the two approaches, and we categorize protocols based on their transmission initiation strategy.

3.3.1 Sender-Initiated

CSMA [24] was the first sender-initiated protocol, with the sender sensing the channel before transmitting. Many protocols that followed, such as MACA [28] and IEEE 802.11 DCF [5], adopted the RTS/CTS mechanism to overcome the hidden node problem in CSMA. The RTS/CTS handshake became a basis for many proposed protocols to come. This handshake is based on the assumption that senders should be the proactive entity in establishing communication, by indicating their intent to transmit data. This handshake was also adopted in many multiple channel protocols to contend for and reserve available channels. For example, PRMA [27], FPRP [32], and MC MAC [51] use a sender-initiated handshake on a common control channel to attempt to reserve a particular data channel. Sender-initiated approaches are more intuitive and more suited to generalized networks with unpredictable traffic patterns.

3.3.2 Receiver-Initiated

In this class of protocols, receivers poll their neighbors with a RTR message, which indicates a node's readiness to receive data. The first such protocol is MACA-BI [31], which was primitive in nature. In MACA-BI, a node sends RTR to its neighbors whenever it is ready to receive data. A polled node that has data to send can subsequently transmit to any node, not necessarily the polling node. MACA-BI is suitable for certain sensor network applications that require sending data in certain directions regardless of node identities.

RIMA-SP [38] adds the restriction to MACA-BI that nodes can only send to the polling node, and that both nodes can send data once the handshake is complete. In RICH-DP [43], receivers use RTR to poll neighbors, but they do so to reserve the current frequency hop. RICH-DP also adds an ACK message that is issued by the receiver when it successfully receives a portion of data. Finally, Lal [49], combines SDMA with a receiver-initiated approach, where a receiver sends RTR omnidirectionally and awaits directional RTS messages from potential senders.

A receiver-initiated MAC protocol yields better network performance for a specific class of ad hoc networks. In sensor networks, for example, the goal is to get data to a certain data sink. The particular source node of the data may not be important, as long as the data is from a certain region of the network. In this case, having a receiver poll its nodes for any available data is desirable. In networks where nodes have data to send often, a receiver-initiated approach also performs well since most RTR messages serve a useful purpose. This approach also has its shortcomings. By merely announcing its readiness to receive data, a receiver does not ensure that exactly one of the neighboring nodes will attempt to send data. Therefore, more recent protocols propose additional mechanisms such as frequency hop reservation or directional RTS messages to mitigate this problem.

3.3.3 Transmission Initiation Summary

The appropriate transmission initiation strategy of a protocol is highly dependent on the potential application areas of that protocol. For generalized networks, a sender-initiated protocol is more suitable. For some specialized networks, such as sensor networks, receiver-initiated protocols are a better choice.

3.4 Topology

Although topology is primarily network layer aspect, certain protocols use topology information for making better decisions at the MAC. Exploiting topology-related information at the MAC layer is an example in which cross-layer design can enhance performance.

Ad hoc and sensor networks may include nodes with varying capabilities and resources. Nodes may also be mobile, so the topology of an active network could change frequently. Therefore, an efficient protocol is one that assumes a topology as generalized as possible. The network must also be able to adapt to heterogeneous node capabilities in a way that optimizes performance and minimizes energy consumption. Network topology can typically be described in terms of hierarchy and hops. A network could have a centralized, clustered, or flat topology. In the centralized case, a single node or base station controls and manages all other nodes in the network. Clustered topologies designate one node in each group of nodes to handle localized central control of the group. Flat topologies consider a fully distributed approach, where all nodes are both nodes and routers, and the notion of centralized control is absent.

We make a further characterization of protocol topologies by examining the hop nature of a network. Some protocols assume that nodes only need to communicate with reachable neighbors, and are referred to as single hop protocols. Other protocols assume that nodes need to communicate beyond their reachable neighbors, and that sometimes a packet has to be relayed through many intermediate nodes to get to its destination. We refer to these protocols as multihop. Single hop protocols are simple but restrictive since they offer limited support for larger networks. Multihop protocols are more general in scope and more scalable, although they introduce added complexity into channel access mechanisms.

There are several combinations of hierarchy and hops in proposed protocols: (A) single hop flat topology; (B) multiple hop flat topology; (C) clustered topology; and (D) centralized topology. This section categorizes protocols according to their topology and explores how each topology choice impacts a network's performance.

3.4.1 Single Hop Flat Topology

Single Hop Flat Topology protocols are not concerned with handling relaying of data, and consider that all nodes are similar in capabilities. Some protocols [24,28,30,31,38,56] assume that a node has a global view of the network, or that higher layer protocols can handle reaching distant nodes. Thus, each node only has to contend with its immediate neighbors in order to obtain access of a channel. As the node density and network load increase in a network, these protocols scale poorly since delay increases exponentially and throughput drops drastically. Other protocols [5,34,35,55] attempt to add scalability and adaptability to harsh network conditions to this topology class, either by providing multiple channels to support an increased number of users [5,55], or by offering techniques such as window splitting and hop reservation [34,35,43] to make better use of the common medium.

In dense or highly loaded networks, however, capacity and performance are limited in a single hop topology, even when using optimization techniques. Other drawbacks of a single hop topology are high power consumption and lack of flexibility. For instance, when two nodes move away from each other, both must increase their transmit power if they still need to communicate. Even if transmit power adaptation is ensured, the price for mobility is increased average power consumption, which could have been avoided in a multihop topology. Therefore, this topology is more suited for wired networks or smaller scale and lower throughput wireless networks, such as Personal Area Networks (PAN) or sparse Local Area Networks (LAN).

3.4.2 Multiple Hop Flat Topology

Multiple Hop Flat Topology protocols offer a more scalable and general approach. Protocols that use this topology also consider that the network is homogeneous in terms of node capability and functionality. Ad hoc and sensor networks can benefit from a multihop topology mainly in power efficiency and correct channel reservation mechanisms beyond one-hop neighbors.

PAMAS [96] addresses the possibility of some nodes being out of range of the receiver or transmitter, and it allows nodes within range of an active transmission to turn off their radios for power efficiency.

Some multiple channel protocols [36,51] also exploit a multihop flat topology to allow for spatial reuse of channels within the network, thus increasing the total number of allowable hosts. Had they been implemented in a single hop topology, these protocols would have only been able to support as many nodes as there are frequency channels.

Other protocols [32,39,40,44,51,190] consider that nodes can only reach their immediate neighbors, but they also have to establish reservations that do not conflict within their two-hop neighborhood. In a single hop topology, channel reservation in these protocols would have to be unique among all nodes, thus preventing any channel reuse.

Some multiple hop flat topology protocols [45,48] not only make use of multihop for channel access, but also for limiting the overall power consumption by forcing nodes to transmit only with the power necessary to reach the receiver. Because of their multihop topology, nodes may choose to send data through a multihop path to a node that is within transmission range [2]. These protocols show how a multihop topology can reduce overall power consumption in the network. Through the use of directional antennas, MAC protocols [49, 54] can direct the emitted power toward the receiver, enabling more active multihop links to coexist on the same channel.

3.4.3 Clustered Topology

One major challenge in ad hoc networks is performing network initiation, management and control. In infrastructure networks, a base station handles these functions. Researchers have found the idea of centralized control attractive, and have tried to emulate it in ad hoc and sensor network models by designing protocols that use a clustered topology. In a clustered topology, one node among each group of nodes is selected to act as the cluster head. Clustered topology protocols attempt to reduce control overhead at individual nodes by placing much of this burden at the cluster head. There are several approaches to choosing the cluster head.

In small sensor networks, the choice of a cluster head is static. Typically, one node, known as the gateway node, has the capability of communicating with the sensor nodes as well as a larger network such as the Internet. If all the nodes are within range of the gateway node, then the cluster head is always the gateway node.

In other networks, the selection of a cluster head can be dynamic. The Virtual Base Station (VBS) [47] protocol focuses on a dynamic random selection of a virtual base station for clusters in mobile networks. In VBS, each node announces its IP address periodically by sending "hello" messages to its neighbors. Nodes listen to "hello" messages and join the cluster of the lowest IP address node in their vicinity. Changes in the network such as nodes moving or dying are adapted to through the periodicity of "hello" messages. Although VBS protocol provides a simple mechanism for choosing a unique cluster head in a specific area in the network, the selection of a cluster head is totally random (since IP addresses do not hold any information relating to nodes). Furthermore, VBS has the undesirable feature that nodes with lower IP addresses suffer from resource overuse.

TDMA-based sensor network MAC protocols [58, 190] also implement the virtual cluster model in order to enable nodes to keep their clocks synchronized. By electing cluster heads periodically and listening to cluster head beacons, nodes can compensate any local clock skews.

An approach that addresses some of the shortcomings of virtual base stations is the Weighted Clustering Algorithm (WCA) [52]. WCA algorithm proposes electing cluster heads based on a weight function at each node that expresses a node's suitability for being a cluster head. In WCA, nodes first

---

[2] [175] describes techniques for deciding when to use a multihop path instead of a single hop path through power considerations.

discover their neighbors, including node degrees, distances and velocities. A node's degree is the number of neighbors that it has. After this initial discovery phase, each node calculates and announces its own weight value. The parameters that contribute to the weight function are: the distance from neighbors (for transmit power consideration); the time a node will spend as a cluster head (battery power available); mobility; and connectivity. The algorithm also proposes that cluster heads of different clusters maintain connections to each other using dual power radios. Finally, WCA supports mobile nodes by allowing handovers for nodes moving from one cluster to another.

Similar protocols [41,46] propose electing a cluster head according to battery power only, which is a simpler but less generalized approach than WCA. Electing cluster heads based on battery power is suitable for large sensor networks, where energy consumption is a critical issue because of the limited energy resources at each node. Nodes that are far from the gateway node can thus elect a cluster head based on battery power.

Bluetooth [6] also has a clustered topology in the form of piconets of one master and up to seven active slaves. The master is not dynamically chosen, rather it is always the initiator or founder of the piconet. Since Bluetooth was designed for Personal Area Networks (PAN), which aim at achieving wireless connections within an office or home, such a selection of the master makes sense. A PC or another central device generally has to be powered on, and that device polls less intelligent devices to establish connections. For other ad hoc networks with more general applications, statically assigning a master is inefficient.

HIPERLAN [18] is another standard that adopts a clustered topology. In this protocol, some nodes are designated as forwarders, which are responsible for relaying data to distant nodes. HIPERLAN designates other nodes as P-supporter nodes to keep track of the sleep schedule of neighboring nodes (see Section 3.5.2). Therefore, in HIPERLAN, forwarders and P-supporters share the duties of a cluster head.

3.4.4 Centralized Topology

Centralized topology protocols [25, 27, 29] require the presence of a central base station to coordinate medium access. For instance, in BTMA, all nodes communicate through the base station, which sends an out-of-band busy signal whenever the data channel is busy to prevent collisions. In PRMA, nodes contend for available time slots in the next frame. A central base station determines the status of each slot in the next frame and announces the successful reservations for the upcoming frame. Finally, MACAW assumes that there are several fixed base stations connected with a wired network.

Ad hoc and sensor networks generally do not adopt a centralized topology, since by definition they are infrastructureless. However, these protocols provide valuable concepts such as a busy tone channel and time slot reservation which are extendable to ad hoc and sensor networks.

3.4.5 Topology Summary

In short, multihop flat topology or a clustered topology are more suitable to ensure scalability in ad hoc and sensor networks. Both of these topologies require more control messaging. In homogeneous networks, a multiple hop flat topology is more appropriate for balancing network loads. In heterogeneous networks, a clustered topology allows the high power nodes to become cluster heads and handle most of the overhead control messaging. A single hop topology requires fewer control messages but it is not scalable. A centralized topology is by definition an infrastructure network, and thus it is not an option for ad hoc networks.

3.5 Power

A major design consideration for ad hoc and sensor network MAC protocols is the power consumption of individual nodes, and the overall power consumption of the network. Power consumption at the nodes stems from data transmission and reception, idle listening to the channel, processing, sleeping, sensing, or actuating. Power consumption is an inherently cross-layer aspect, since some of the above power sinks, such as processing and sensing, are not MAC layer functions. This section reveals how ad hoc and sensor network MAC protocols can exploit cross-layer power-related information for optimizing performance.

Power conservation is important for any type of mobile node, whether operating in an ad hoc or infrastructure network, because of its limited battery power. In infrastructure networks, a resourceful base station is responsible for managing channel access and allocation, while nodes consume most of their power for data transmissions. In ad hoc and sensor networks, however, the absence of a base station places the burden of control on one or more of the nodes. Furthermore, the absence of a centralized controller increases chances of collisions and channel assignment conflicts that lead to higher power consumption in the form of control signaling and retransmissions. Sensor networks are particularly sensitive to the issue of power consumption because nodes must endure long deployment with the limited available energy resources.

Because the medium access strategy significantly impacts energy consumption of a mobile node, we can achieve much of the power optimization in ad hoc and sensor networks through careful design of the MAC protocol. In this section, we describe common mechanisms for power conservation, and we classify protocols based on the mechanisms they adopt.

3.5.1 Transmit Power Control

A major cause of power consumption at a node is transmission power, a physical layer parameter. MAC protocols that operate on transmission power adopt a form of cross-layer interaction between the MAC and physical layers. Some protocols [45,46] have proposed controlling the transmit power so that it is just enough to reach the intended receiver. For example, DCA-PC specifies that each node continuously monitors, records and updates the transmission power level it needs to reach each neighbor. In DCA-PC, a node is initially unaware of the appropriate power levels, so it transmits with maximum power. Once it establishes contact with a neighbor, then both nodes learn the appropriate power levels they need to communicate.

DPC/ALP [48] is another protocol that supports transmission power control. Recall that in DPC/ALP, a node sending its RTS progressively increases its transmit power until it exceeds a threshold of detection at the receiver. If the receiver replies, then a connection is established, otherwise the sender backs off. During data transmissions, a sender in DPC/ALP also transmits at the minimum power needed to overcome noise at the receiver.

Lal [49] and MMAC [54] exercise another method of power control through directional antennas. Because nodes send messages in the direction of the intended receiver, the transmission requires less power than in the omnidirectional case, where the signal is scattered in all directions.

Some protocols propose power control enhancements to the IEEE 802.11 MAC protocol [154, 234]. These protocols specify that a sender and receiver transmit RTS and CTS control messages at maximum power so that neighbors become aware of the upcoming transmission. The sender can subsequently transmit the data at a lower power level which is directly related to the distance between the pair of nodes, instead of the maximum power level. The work in [233] has shown that this approach may produce asynchronous links and that it may lead to collision in the carrier sensing zone of the sender. Reference [233] proposes that the sender and receiver periodically raise the power level during the sending of data to keep neighbors within the carrier sensing zone aware of the ongoing transmission.

Transmission power control benefits dense or highly loaded networks, where a large number of nodes need to efficiently share the wireless medium with minimal interference. Protocols that support this mechanism must also use few data channels to avoid overuse of the common control channel.

3.5.2 Sleep Mode

Some protocols acknowledge that in ad hoc and sensor networks, a considerable portion of power consumption is wasted due to overhearing irrelevant transmission, or due to idle listening to the channel. PAMAS [96] takes advantage of the simple RTS/CTS handshake to avoid this problem. As mentioned earlier, PAMAS has a common control channel and a common data channel. Nodes that hear RTS or CTS on the control channel refrain from communicating since they are in the neighborhood of either the sender, receiver, or both. These neighboring nodes power off their transceivers for the duration of the transmission indicated in the handshake messages. Therefore, PAMAS reduces battery power consumption in highly connected low load networks, where at any time, many idle nodes overhear other nodes' transmissions.

HIPERLAN [18] also allows nodes to go into sleep mode to conserve power. Such nodes are called p-savers, and must set up a specific wake-up pattern by notifying specialized neighboring nodes called p-supporters. P-supporters are responsible for keeping track of the sleep schedules of neighboring p-savers, for buffering data for these nodes, and for forwarding it to them when they are set to wake up. This mechanism obviously requires extra buffer space and battery resources at p-supporter nodes.

Bluetooth [6] supports three low-power states: Park, Hold, and Sniff. Park state provides the lowest duty cycle and thus the lowest energy consumption. In this state, a node releases its MAC address but remains synchronized with the piconet. The node wakes up occasionally to synchronize and listen for broadcast messages. Hold state is the next higher low-power state. In Hold state, a node keeps its MAC address and transmits immediately after waking up. Finally, in the Sniff state, a node listens to the piconet more often than in the Hold state, but still at a lower rate than normal. The rate at which a node listens is programmable and application-dependant.

Sleep modes are especially useful in sensor networks. Sensor network deployments could span several months or years. For much of that deployment time, the nodes wait for events to happen in the environment and they do not transmit any data. Consequently, nodes should power down their radios to avoid wasting energy for idly listening to the channel. BMAC [56] enforces sleep modes through the definition of asynchronous check intervals. BMAC empowers each node to set the frequency at which to wake up its radio to check for channel activity. Other sensor network MAC protocols [58,190] rely on synchronization and snooping on neighbors' control messages for determining wake-up times.

The use of sleep modes involves an inherent tradeoff between energy consumption and delay in ad hoc and sensor networks. Nodes that wake up less often save energy, but they also incur more delay for data delivery. Long-term sensor network monitoring applications favor this tradeoff, since energy consumption is the main concern in these applications. On the other hand, this tradeoff may not be suitable for real-time applications, such as voice or video transfer through an ad hoc network. Another potential drawback of supporting sleep modes is the overhead power consumption for powering up and powering down a transceiver. In some cases, this overhead may exceed the power savings of supporting sleep mode. Therefore, whether it is beneficial to support sleep mode depends on the specifications of particular transceivers.

3.5.3 Battery Level Awareness

There are several protocols that are aware of battery power levels at nodes and adjust their behavior accordingly. A common strategy is to base the selection of a cluster head on battery levels. DPC/ALP and Jin also classify nodes into High Power (HP) and Low Power (LP) according to remaining battery power. DPC/ALP gives LP nodes priority during transmissions, by allowing them to reserve slots sooner than HP nodes. All of these protocols produce power savings when they are used in a power heterogeneous network. For example, a network may include laptops, palmtops, and pens equipped with transceivers. One of the laptops would generally be selected as a cluster head because of its relatively high power resources. For similar reasons, electronic pens get priority for transmission in DPC/ALP due to their limited battery power.

WCA [52] also considers battery power along with several other parameters to elect a cluster head. Because it combines the effects of several factors in electing a cluster head, WCA performs well in both power homogeneous or heterogeneous networks.

A form of battery level awareness can also promote power savings in sensor networks. Although the battery level of a sensor node is a critical factor, measuring the absolute battery level of a sensor node is inaccurate because of fluctuations in voltage. Instead, sensor nodes can make use of a relative indicator of battery level, such as the node's radio duty cycle during the past time window [192]. Nodes could use this relative measure of battery level for adapting MAC behavior, such as setting the radio duty cycle dynamically.

3.5.4 Reduced Control Overhead

The exchange of control messages prior to data transmission is also a source of power waste. Whereas control messages are necessary to avoid collisions, reducing these messages to the minimum is beneficial. MARCH [42] is one protocol that follows this reasoning. As described earlier, in a path where there are N hops, MARCH uses 1 RTS message and N CTS messages. When N is large, power savings from this approach are considerable. However, MARCH ignores the case of heterogeneous power nodes. Referring to Fig. 3.4, when a node A has data to send to another node Z, using the path A-B-C-D-Z, A sends RTS_A to node B. B replies with CTS_B to indicate to A that it is ready to receive data. MARCH assumes that the next node in the path C hears CTS_B when it was sent to A. In this assumption, MARCH supposes that all nodes are equidistant in the network, and that nodes are always transmitting at a constant power level. If transmit power control was used in conjunction with MARCH, then the mechanism does not work. For example, C might be further away from B than A, in which case C does not hear CTS_B.

3.5.5 Savings for Particular Settings

Given that different networks have varying particularities when it comes to power, some protocols focus on achieving power savings for specific settings. One of these protocols, SRMA/PA [44], is concerned with quality of service and considers that there are high and low priority traffic. It allows high priority traffic to preempt the low priority traffic when trying to grab slots for transmission. Markowski's [34] protocol follows a similar reasoning for three traffic classes. This technique improves performance for high priority traffic, but it has the opposite effect for lower priority traffic. In a network with many voice or real-time traffic nodes, both protocols reduce overall network power consumption by avoiding collisions and retransmissions for real-time traffic.

3.5.6 Increased Control Overhead

There are some protocols that are strictly separated from power issues, and therefore do not incorporate power considerations into their behavior. A few of these protocols, however, contain power wasting features. For example, the approach of RICH-DP [43] is suitable for networks with predictable and periodic traffic. In networks that do not fit this description, many RTR messages are sent while no nodes have data to send. In such networks, RTR messages present two causes of power waste. The first is the transmit power of the node sending a non-useful RTR. The second source of power waste is due to idle listening to RTR messages at neighboring nodes.

TDMA-based protocols also contain a regular source of power waste to maintain synchronization. Some of these protocols [32, 55] assume that nodes have a GPS radio to maintain synchronization. Although GPS is effective and reliable for keeping nodes synchronized, GPS radios consume valuable battery power resources at each node when periodically receiving synchronization messages. Nodes using a CSMA [24] protocol also waste power through idle listening to a busy channel.

In all multiple channel protocols with a flat topology, channels are assigned dynamically at each node. This represents control overhead and therefore wasted power for performing channel assignments. In these protocols, nodes typically need to monitor different channels for availability, or to adopt a greedy approach in grabbing channels. Monitoring channels dictates that the node's transceiver is frequently active thus wasting more power. In a greedy approach, the chance of conflicts and collisions is increased since a node tries to grab channels that may already be used. A clustered topology typically reduces the chance of conflicts by limiting the number of nodes contending for channels and by assigning a portion of the control tasks to a cluster head.

3.5.7 Power Summary

The extent to which each of the different power saving mechanisms in this section actually conserves power is dependent on the application scenario. A multi-purpose protocol should enable the selection and tuning of a wide variety of power-saving mechanisms for different application scenarios, without adding overhead that would counterbalance the benefits of having these features.

3.6 Traffic Load and Scalability

A majority of the surveyed protocols perform well for their intended applications. In this section, we examine the quality of service classes served by the available MAC protocols: (A) high load; (B) high density; (C) real-time traffic; (D) long-term operation and (E) more selective scenarios. For each traffic load or type, we assess the scalability and adaptability of protocols to dynamic network conditions. Our discussion of performance is qualitative, based on parameters such as channel utilization, throughput, and delay. Note that quality of service is an inherently cross-layer aspect that incorporates indicators from several layers.

3.6.1 Highly Loaded Networks

Receiver-initiated approaches operate well in networks where channel utilization is high. Since nodes send RTR messages whenever they are ready to receive data, there is a high probability that whenever a sender has data, it can find an appropriate receiver to relay its data. RICH-DP and Lal's SDMA protocol both reserve channels (one does so in space and the other does it in frequency) for data transmission once the handshake is complete. As a result, these two protocols exhibit reduced collisions and increased efficiency. Both of these protocols are highly adaptive to varying network conditions, and both offer mobility support. Lal's SDMA approach is especially effective at bottleneck nodes, where support for simultaneous multiple transmissions is desirable.

PS-DCC [37] also adapts well to high network load. Because it always calculates the sending probability based on current channel utilization, PS-DCC reduces collisions in the network when utilization is high by forcing individual nodes to wait for longer durations before transmitting. This technique clearly introduces increased transmission latency. If it is used with other channel separation mechanisms, this scheme becomes attractive in its simplicity and effectiveness. PS-DCC is also adaptive to the general network case, where nodes are moving around and topology changes are frequent.

GRID-B [50] is designed to manage areas in the network where the load is high, which are referred to as hot spots. By adaptively borrowing channels from a neighboring area, a hot spot can support the required load. ADAPT [39] is another protocol designed to handle high load networks, since it allocates slots to each node statically. Borrowing of unused slots allows nodes with a high traffic rate to have increased access to the channel. Neither GRID-B nor ADAPT are adaptive to a rapidly changing network topology or to highly mobile nodes due to their static assignments of resources. In GRID-B, high mobility might bring nodes out of the predefined geographic area for which the channels were initially assigned. Similarly in ADAPT, each slot has a predefined owner. New nodes that enter the network do not own any slot and thus have a reduced priority, since they are only allowed to contend for other nodes' slots.

TDMA protocols (see Sect. 3.2.2) are generally adaptive to highly loaded networks with periodic traffic, such as a voice-dominated network. These protocols, however, do not cope well with random data traffic, and are not scalable, since increased network size requires more nodes to contend for a fixed number of slots.

3.6.2 Dense Networks

Protocols that perform best for dense networks base their behavior on power considerations. Through transmission power control, GPC [46] and DCA-PC [45] limit the possibility of collisions among nodes. DCA-PC performs well in high mobility situations, as well as for larger networks. The only constraint on DCA-PC is to use few data channels in each vicinity and few power levels to avoid control channel overuse. As for GPC, it dynamically chooses forwarding agents based on battery power level, which avoids overusing the resources of a single node and thus promotes fairness in a dense network.

MAC protocols that use directional antennas [49, 54] also perform well in dense networks. Although these protocols are appropriate for both high load and dense networks, which seems attractive, some issues about the economic feasibility of SDMA remain unresolved. For example, the use of directional antennas for sensor networks is questionable because of the hardware complexity and cost of directional antennas. GRID-B handles dense networks through its hot spot mechanism in the same way it adapts to high load situations.

MCSMA [36] also offers a solution for dense but lightly loaded networks through its use of several frequency channels that are spatially reused within small distances. In a dense MCSMA network, the per node bandwidth is reduced but the overall channel utilization of the network is increased.

3.6.3 Voice and Real-Time Traffic

Some protocols are more suited for voice and real-time traffic. These protocols typically have 2 common attributes: priorities and reservations. To support priorities, a protocol must classify nodes or traffic into two or more classes. Each class typically has a certain priority level based on node features and nature of traffic. Reservations are usually allowed for higher priority traffic. PRMA [27] was designed to support voice communication in a data network over the wireless medium. By allowing voice nodes to reserve slots for subsequent frames, PRMA ensures that once a voice node reserves a slot, it is guaranteed the needed bandwidth to maintain an acceptable quality of service. This protocol offers improved performance over pure TDMA, but it assumes there is a base station to maintain synchronization and resolve contentions. An improved adaptation of PRMA for ad hoc networks could be achieved by implementing it within a clustered topology, where a cluster head performs most base station functions. D-PRMA [55] adds a fully distributed flavor to the original PRMA, by having nodes resolve contention for slots among themselves. D-PRMA performs well for periodic traffic networks. SRMA/PA [44] follows a similar approach to PRMA, by designating nodes as voice and data terminals. Since it allows voice nodes to preempt data nodes, SRMA/PA favors voice nodes even more than PRMA. SRMA/PA also performs well in a network with many voice terminals.

VBS [47] is another protocol designed for voice and real-time traffic. After it is elected in a cluster, the VBS allocates virtual circuits to nodes that request connections. Virtual circuits ensure a certain bandwidth allocation, which makes them suitable for supporting real-time traffic. The VBS protocol exhibits stable performance when it comes to VBS changes and cluster memberships. It also provides a mechanism for handovers between neighboring clusters which supports mobility. However, this protocol focuses mainly on managing topology and clusters, and does not elaborate sufficiently on data transmission and related issues.

Markowski's window splitting protocol [34] is also favorable towards real-time traffic, by allowing hard real-time nodes to preempt soft real-time nodes that also preempt non real-time nodes. Because of its fully distributed nature, the window splitting scheme can theoretically handle a large number of nodes of each traffic class. An increased number of senders, however, causes the active window to become too small, and that in turn causes increased transmission latency for many nodes. Thus, this protocol performs well in real-time traffic networks that are sparse and limited in size. It can also enforce a firm real-time traffic class, where a node attempts to send a packet for a few times as soft real-time, and if those attempts are unsuccessful, then it sends the packet as hard real-time.

3.6.4 Unattended Long-Term Operation

Sensor network deployment could last for several months or even years. As such, unattended long-term network operation is a major goal for many sensor network MAC protocols. The main challenge in supporting unattended long-term operation is making efficient use of the limited energy resources at each node. BMAC, SMAC, and TMAC all try to efficiently manage the radio duty cycle at each node in order to minimize energy consumption at each node. All three protocols acknowledge that idle listening on the radio channel contributes significantly to node power consumption. As such, these protocols propose techniques for reducing idle listening by placing radios into low-power states when there is no data to transmit or receive.

3.6.5 More Selective Scenarios

There is a group of protocols that performs best when used in more specific situations, other than traffic deadline restrictions. For example, HRMA [35] performs well when packet sizes are large. Its performance degrades, however, when node density increases. In Jin's proposed protocol [41], low power nodes get priority over high-power nodes. In a dense network with a large percentage of low-power nodes, high-power nodes may experience large transmission delays. However, if a network only has a few low-power nodes, high-power nodes experience tolerable transmission delays.

MC MAC [51] also has specific constraints for adequate performance. Although the number of usable codes is around 30, MC MAC proposes an optimal performance for a 7 code network, to avoid overusing the control channel. The protocol also performs best for short range applications. In MC MAC, a highly loaded network also causes a high access delay.

Ad hoc and sensor networks require versatile and flexible MAC protocols, so protocols that perform well for more selective scenarios are unattractive for widespread adoption in general ad hoc and sensor networks.

3.6.6 Traffic Load and Scalability Summary

In short, multiple channel protocols and power-efficient protocols exhibit better performance for high load and high density networks. TDMA and reservation-based protocols perform best for networks dominated by voice and real-time traffic. Long-term sensor networks benefit from MAC protocols that efficiently manage the radio duty cycle. Protocols that perform well for selective scenarios are not suitable for a general ad hoc network.

3.7 Logical Link Control

The Logical Link Control (LLC) layer lies above the MAC layer and is responsible for performing three main functions:

1. Managing frames to upper and lower layer: LLC provides an interface between the MAC sublayer and layer 3, the network layer, by performing packet encapsulation and decapsulation.
2. Error control: LLC enforces error control through mechanisms such as checksums and parity bits. The two forms of possible errors are bit errors and frame errors. Bit errors indicate errors in a number of bits in a frame. Frame errors indicate badly synchronized or badly received frames.
3. Flow control: LLC defines the data values used in flow control signaling between two transmitting hosts, although the actual flow control occurs at layer 4.

While managing frames to upper and lower layers happens internally to a node, both error control and flow control involve interaction with other nodes. Many protocols retransmit a frame whenever the receiver detects an error. While retransmission increases a network's reliability, it also may have adverse effects on channel utilization, energy consumption, and delay in the network, particularly on higher layers in the network stack.

The following example demonstrates these adverse effects. A network that uses the Transmission Control Protocol (TCP) [109] usually enforces congestion control at layer 4 in order to deal with network congestion. A sender monitors a receiver's acknowledgement messages in order to adjust its transmission rate. If the sender detects a congestion in the network because of delayed acknowledgements, the sender drops its transmission rate by half to help reduce congestion. In many cases in wireless networks, the delayed acknowledgements may arise because of transmission errors rather than congestion. The sender in this case would still drop the transmission rate to alleviate congestion.

This example shows one of the drawbacks of strict layering. Because the transport layer is unaware of the cause of the delayed acknowledgements, it always assumes the cause is congestion. Cross-layering could help mitigate this issue by enabling the transport and LLC layers to share information. Whenever the transport layer misses receiver acknowledgements, it checks the LLC state to decide whether the missed acknowledgements are due to errors or congestion. Subsequently, the transport layer could make the right flow control decision.

3.8 Conclusion and Discussion

Ad hoc and sensor networks provide a distributed communications paradigm that can be extended to fit into the "anytime anywhere" concept of ubiquitous computing [23]. One major obstacle that impedes the proliferation of such networks is the tight regulation exercised on unlicensed communications, restricting frequency bands and bandwidth where ad hoc networks may be used. Whether this distributed model will be applied to a broader range of networks and become dominant in the coming years remains to be seen. This will also depend in part on regulatory issues. The other challenge towards the development of ad hoc networks is the design of efficient self-management protocols. In our survey of ad hoc and sensor network MAC protocols, it has become evident that the overwhelming majority of these protocols were derived heuristically and were aimed at optimizing a particular set of measures under a particular set of operating conditions. However, most of these heuristics lacked generality and were not tested in a deployed network. Establishing a principled framework for optimizing ad hoc and sensor network behavior is challenging since there is clearly a wide range of applications and potential physical layer technologies that have different considerations.

4

Network Layer

The third layer of the OSI model is the network layer, shown in Fig. 4.1. The network layer is responsible for all of the aspects of end-to-end packet delivery, including logical message addressing and routing packets between different networks. The main goal of a routing strategy is to efficiently deliver data all the way from the source to the destination. Although all routing protocols share this goal, each protocol adopts a different approach to achieve it. The routing strategy has a significant impact on the performance of ad hoc and sensor networks, especially since the nodes act as routers.

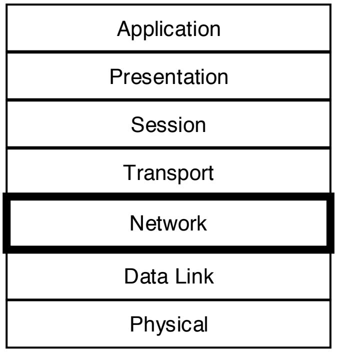


Fig. 4.1. The network layer in the OSI reference model

First, routing protocols for ad hoc and sensor networks have to be robust to the unreliable wireless links in ad hoc networks, which are due to interference variations and mobility. Second, node mobility introduces another degree of complexity over wired or infrastructure networks, especially because of the lack of a central network controller. Third, energy awareness is crucial in ad hoc and sensor network routing protocols. Because the nodes also act as routers in ad hoc and sensor networks, energy depletion of some nodes could mean loss of connectivity in the network.

A characterizing feature of routing protocols is the manner in which they disseminate routing state among the nodes. The topology of a routing protocol also impacts performance related to energy efficiency, delay, and throughput. For example, a clustered topology protocol may save energy at most nodes while incurring more delay for cluster formation. Routing protocols also adopt either a connection-oriented approach or a connectionless approach. Connection-oriented communication, such as phone communication, establishes a predefined path for communication between two nodes, similar to trains in the transportation industry. Connectionless communication, such as Internet traffic, routes data on a hop-by-hop basis, resembling the decisions that car drivers make at each intersection between the source and destination.

In addition, cross-layer design enables ad hoc and sensor network routing protocols to exploit information from other layers, providing the following features:

- **Power-Awareness**: Efficient power management at ad hoc and sensor networks' nodes requires power-aware mechanisms at every layer. Power-aware routing protocols incorporate energy-related metrics into routing cost computation and path selection.
- **Location-Awareness**: With the availability of positioning systems such as GPS, location-awareness is an integral part of many routing protocols for ad hoc and sensor networks. In most cases, location information enables routing protocols to make quicker and more focused decisions for establishing optimal paths.
- **Quality-of-Service**: There is a wide range of quality-of-service requirements for ad hoc and sensor networks, including throughput, delay, reliability, or energy consumption. Most routing protocols involve tradeoffs between these QoS metrics. As a result, several routing protocols for ad hoc and sensor networks adopt flexible mechanisms for tuning QoS variables to support various applications.

This chapter is not intended as a comprehensive survey of existing routing protocols for ad hoc and sensor networks. Instead, we refer to existing routing protocols to uncover the strengths and weaknesses of routing protocol design choices for ad hoc and sensor networks. For comprehensive reviews on routing protocols for ad hoc and sensor networks, we suggest the surveys in [59,60,181].

The remainder of the chapter is structured as follows. Section 4.1 distinguishes the different classes of routing protocols according to their state dissemination policy. Section 4.2 explores the effect of topology choices on the performance of routing protocols. A related topological choice is whether to use multipath routes, which we discuss in Section 4.3. Sections 4.4 through 4.6 examine proposed routing enhancements for ad hoc and sensor networks that cut through the traditional layered approach by using information from other layers. The enhancements include power-awareness (Section 4.4), geographical routing (Section 4.5), and quality-of-service (QoS) (Section 4.6).

4.1 Route State Dissemination

The routing state indicates the state of links and nodes in the network. Most routing protocols incorporate information on the routing state for determining how to forward packets.

Primitive forwarding methods do not rely on any route state. One of the simplest protocols is flooding, where nodes broadcast their packets to all neighbors. Neighbors then rebroadcast the packets and this process is repeated until the packet reaches the destination . Obviously, flooding does not store any routing state at nodes, but it involves a lot of overhead and it may lead to implosions in the network.

An improved version of flooding is gossiping, in which each node sends data packets to a randomly selected neighbor. Upon receiving a data packet, a node checks if it is the intended destination of the packet. If not, then the node again forwards the received packet to another randomly selected neighbor, and this process goes on until the packets reaches the destination. Although gossiping involves fewer packet transmissions than flooding, its higher delay and resource-blindness make it inefficient for resource-limited nodes.

In this section, we discuss the three different route dissemination strategies that ad hoc and sensor network routing protocols adopt: (1) proactive; (2) reactive; and (3) hybrid.

4.1.1 Proactive Routing Protocols

Proactive routing protocols, which are also referred to as table-driven protocols, maintain a routing state table at each node. Many proactive routing protocols have their roots in Internet routing protocols. Distance vector protocols, such as Routing Information Protocol (RIP) [62] and Internal Gateway Routing Protocol (IGRP) [63], use the hop count to the destination as the routing metric. In both RIP and IGRP, nodes periodically broadcast their entire routing tables to their direct neighbors, enabling each node to maintain a table with the hop count to every known destination in the network.

In contrast, link state protocols enable nodes to flood the network with only the state of local links. As a result, each node can store the state of all the links in the network. Common link state protocols, such as Open Shortest Path First [64], use the state of local links as the main routing metric.

One of the first proactive ad hoc routing protocols is Destination Sequenced Distance Vector (DSDV) [182]. DSDV is based on the Bellman-Ford algorithm for shortest paths.

We use the topology in Fig. 4.2 to illustrate the operation of DSDV. Every node maintains the next hop and distance information to all other nodes in the network. In order to maintain table consistency, DSDV periodically transmits routing table updates. Table 4.1 shows a snapshot of the routing table at node A corresponding to Fig 4.2. For example, if node A has to send data to node E, it uses node B as the next hop on the 2-hop route to node E. The last column in the routing table uses an enhanced sequence number to indicate fresher routes and to avoid routing loops in the network. If node A receives a routing update packet from node B with a sequence number less than 323, then A discards the stale information in the packet to avoid creating a routing loop.

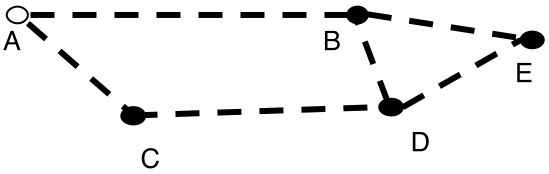


Fig. 4.2. DSDV example topology

Destination	Next	Metric	Seq. Nbr.
A	A	0	A-543
B	B	1	B-323
C	C	1	C-276
D	C	2	D-188
E	B	2	E-206

Table 4.1. Routing table at node A

Optimized Link State Routing (OLSR) [66] is another recent proactive routing protocol for ad hoc networks. Each node using OLSR periodically floods the cost of all the links to which it is connected throughout the network. The availability of all link costs locally at each node enables the node to compute the cost of reaching every other node using shortest path algorithms. An advantage of OLSR over DSDV is its support for unidirectional links.

Proactive routing protocols have also been developed specifically for sensor networks. Because sensor nodes have limited energy resources, proactive routing protocols must limit periodic state updates through selective control. For example, Sensor Protocols for Information via Negotiation (SPIN) [67] is a proactive routing protocol for sensor networks that enables nodes to perform negotiations to eliminate redundant data transmissions in the network. Through SPIN, nodes use metadata to name their data and make communication decisions based on application-specific knowledge of the data. SPIN uses Advertise (ADV) and Request (REQ) messages between a sender and receiver to announce data availability and interest respectively. As such, SPIN adopts a selective version of flooding that is more energy-efficient.

Figure 4.3 illustrates the operation of SPIN. The node A samples its sensors and advertises the availability of sensed data through an ADV message. Upon receiving ADV from A, node B requests the data through a REQ message. Node A then sends the data to node B. Now node B has the data. It repeats the process of advertising that it has the data, and any neighbor of B can obtain the data by sending back a REQ message.

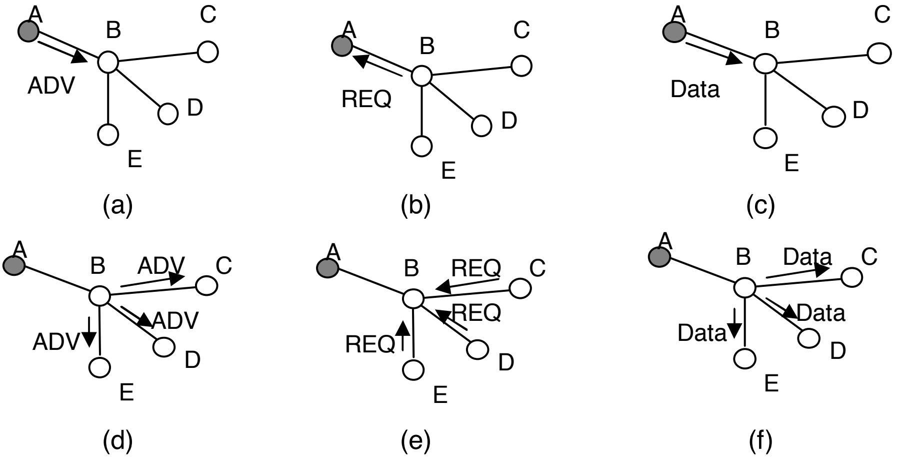


Fig. 4.3. SPIN operation: (a) Node A advertises the availability of data. (b) Node B requests A's data through a REQ message. (c) Node A sends node B the data. (d) (e) (f) cycle repeats between B and its neighbors

Other sensor network routing protocols have also adopted a proactive routing strategy. Low Energy Adaptive Clustering Hierarchy (LEACH) [68] relies on a clustered topology in which nodes periodically and randomly choose whether to be cluster heads. After non-cluster nodes associate into a cluster, cluster heads set a TDMA schedule with one time slot for each node in the cluster.

Based on LEACH mechanisms, Power Efficient Gathering in Sensor Information Systems (PEGASIS) [69] assumes each node has a global knowledge of node positions to construct a chain topology. Nodes in PEGASIS take turns being cluster heads.

Finally, Threshold Sensitive Energy Efficient Sensor Network Protocol (TEEN) [70] uses the same periodic cluster formation method as LEACH, but it also incorporates thresholds at each node to determine whether to report measured sensor values. While TEEN's cluster formation activity is proactive due to its periodic nature, it enables nodes to provide data selectively according to the threshold values.

Proactive routing protocols have low latency. Nodes store up-to-date routing state information locally, so nodes can immediately forward data as it becomes available. Proactive protocols generally involve high overhead cost for maintaining state information, and the overhead cost for periodic state maintenance does not scale well for larger networks. As a result, proactive routing protocols are suitable for smaller networks. Networks with a high traffic load highlight the latency benefits of the persistence of state in proactive routing protocols, which outweighs the overhead for periodic state updates.

4.1.2 Reactive

Reactive routing protocols, also referred to as on-demand protocols, collect state information in response to certain events in the network such as user requests. On-demand state dissemination incurs lower overhead than proactive protocols by eliminating the need for periodic state updates. However, reactive protocols also experience a higher delay since routes must be repeatedly discovered for each data transmission.

Ad Hoc On-Demand Distance Vector Routing (AODV) [71] is a reactive protocol based on DSDV. In AODV, a sender first broadcasts a Route Request Packet (RREQ) with the sender's id and a unique destination sequence number to all its neighbors. All neighbors that receive the RREQ rebroadcast it. Neighbors also store the neighbor's id from which they received the RREQ, which represents the reverse path to the destination. Any node that has already processed this RREQ discards any duplicate RREQs. Finally, when the destination node receives a RREQ, it sends a RREP which eventually reaches the original sender through the reverse path links. The sender then proceeds with data transmission. Note that nodes in AODV maintain only next hop routing state, which provides AODV with a high degree of scalability.

Figure 4.4 illustrates the operation of AODV. In order to reach the destination node E, the sender node A floods its outgoing links with RREQ packets (Fig. 4.4(a)). Nodes B and C receive RREQ from A, and they locally store the id of node A as the next hop on the reverse path for RREP. Nodes B and C also rebroadcast A's RREQ (Fig. 4.4(b)). Upon receiving RREQ from nodes B and C, node A discards these packets since they represent duplicates. Node E receives RREQ from node B, and it records B as the next hop on the reverse path. At this point, node E sends RREP to node B, which then forwards RREP back to node A (Fig. 4.4(c)). Subsequently, data transmission can proceed from node A to E on the established path (A-B-E).

Dynamic Source Routing (DSR) [72] is similar to AODV in its route discovery mechanisms through its use of RREQ and RREP messages. However, DSR specifies that the full source route is aggregated in RREQ packets. The RREP message also contains the full source route. Similarly, each data packet in DSR contains the full source route in its header. The inclusion of the full source route in all messages reduces delay and storage overhead in the network since each intermediate only needs to examine the packet header for

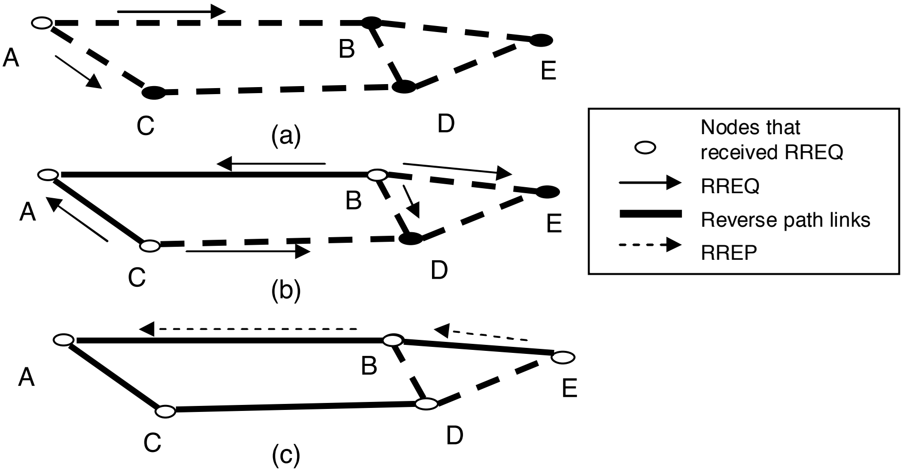


Fig. 4.4. AODV operation: (a) Sender node broadcasts RREQ. (b) Neighbors forward RREQ on outgoing links and maintain reverse path link information. (c) Destination receives RREQ and sends RREP, which is propagated along reverse path to sender

determining the next hop. Only the source node must maintain the full route state information. However, communication overhead increases with DSR, especially as the network size grows, since each packet header contains the full source route.

MIMO Routing Protocol (MIR) [73] proposes a cross-layer variation of the DSR reactive model, coupling the routing and physical layers. MIR relies on Multiple Input Multiple Output (MIMO) technology, which uses smart antennas with multiple elements to provide high spectral efficiencies especially in rich multi-path environments[1]. Nodes can employ the increased spectral efficiency of MIMO technology to improve the link throughput (using spatial multiplexing) or to improve the link signal quality (using diversity gain). The improvement in signal quality serves in reducing the signal bit error rate (BER) or in extending the transmission range. Building on MIMO technology, MIR is a reactive routing protocol that seamlessly adapts between the different MIMO operation modes according to network conditions. In particular, MIR relies on the spatial multiplexing mode as a baseline for static or high density networks. For low density or high mobility networks, range extension is employed. Finally, MIR uses the BER reduction mode for high loss scenarios. While MIR uses the same control messaging sequence as DSR, it extends the range of RREQ messages to several hops through the MIMO range extension mode.

---

[1] MIMO technology has been adopted for recent standardization efforts such as IEEE 802.11n WLAN draft standard [74] and 802.16 (WIMAX) [75].

Temporally Ordered Routing Algorithm (TORA) [76] is another reactive routing protocol that uses a directed acyclic graph rooted at the destination node to route packets. TORA uses a link reversal algorithm to update the directed acyclic graph. If a directed link break leads a non-destination node to have no more outgoing links, the isolated node begins transforming its incoming links into outgoing links. This step is repeated at intermediate nodes until the destination node is the only node with no outgoing links. The link reversal algorithm is distributed and loop-free, but it involves considerable time and message complexity to reach steady state.

Figure 4.5 illustrates the link reversal function of TORA through an example. At steady state, the routing graph is a directed acyclic graph rooted at the destination E. A break in link (D,E) causes link D to have no more outgoing links, leading D to reverse the direction of all of its incoming links. The reversal of link (C,D) causes node C to have no more outgoing links, which leads C to reverse all of its incoming links. TORA reestablishes a directed acyclic after the destination node is the only node with no outgoing links.

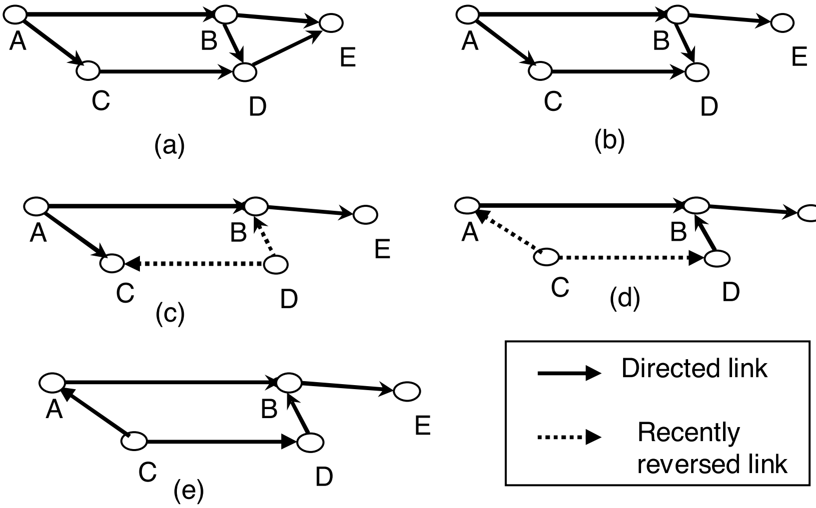


Fig. 4.5. TORA operation: (a) Directed acyclic graph rooted at destination E. (b) Link from D to E breaks. Node D has no more outgoing links. (c) D reverses all its incoming links. Now node C has no more outgoing links. (d) Node C reverses all its incoming links. (e) Directed acyclic graph restored with only node E having no outgoing links

Directed Diffusion (DD) [78] is a query-driven reactive protocol for sensor networks. Sink nodes flood the network with interest messages, which define the sensor events of interest. Interest messages are then diffused through the network, and each node maintains a cache of active interests and the associated gradients. A gradient includes the node to which data should be forwarded, the desired data rate of the interest, and the duration of data forwarding. When an event that matches an active interest occurs, the node that sensed the event sends data in the direction of the gradient. Once the data sink starts receiving data, it reinforces desirable paths from the data source by sending more frequent interests along that path. The sink can also negatively reinforce less desirable paths.

Figure 4.6 provides an example of directed diffusion. The designated sink node A floods interest messages in the network. Each of its neighbors store this interest locally and they forward this interest to further nodes. When an event occurs, node F forwards its data to nodes C, G, and E through which it received the interest from node A. As the data starts arriving at node A, node A begins sending more frequent interest messages along the favorable path through C. Consequently, node F develops a higher gradient for the path F-C-A and uses this path for data transmission. DD combines on-demand route setup, which is energy-efficient, with query flooding, which is not. Overall, the protocol provides energy and delay benefits, but its query-driven structure may not be suitable for continuous data delivery.

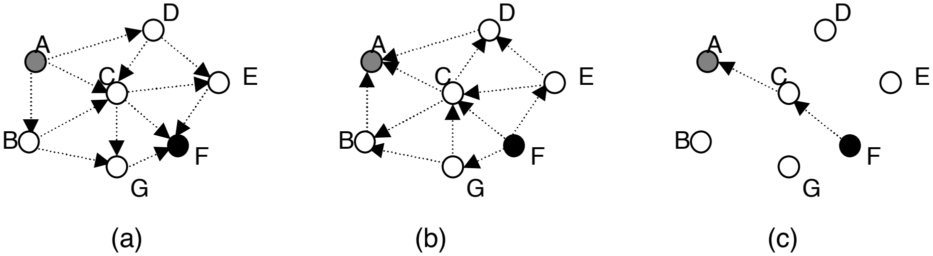


Fig. 4.6. Directed diffusion: (a) Data sink A floods interests in the network. (b) An event occurs at node F, causing F to initially send its data to nodes C, G and E through which it received the interest. (c) Node A reinforces the path F-C-A through more frequent interest messages, causing node F to use this optimal path for sending data.

Reactive protocols involve less communication overhead than proactive protocols since routes are formed on-demand. Thus, reactive protocols scale well and they are suitable for larger networks but they tend to involve high delay for forming on-demand routes. Reactive protocols perform well for low traffic requirements. When the traffic load is high, on-demand path formation becomes less efficient.

4.1.3 Hybrid

Hybrid protocols combine both proactive and reactive strategies in an attempt to get the best of both worlds. Zone Routing Protocol (ZRP) [77] splits the network into zones. For intrazone communications, ZRP uses proactive routing. For interzone communication, it uses reactive routing.

Rumor Routing (RR) [79] is a hybrid protocol that is similar to Directed Diffusion. RR attempts to balance event and query flooding in sensor networks. Whenever an event happens, a node employs an agent, which is a long-lived packet, to traverse the network and notify distant nodes of the event. In essence, the event-based creation of agents in RR is the protocol's reactive aspect. The proactive aspect is the agent traversal of the network, which notifies random - occasionally uninterested - nodes of the occurrence of an event. RR reduces the overhead of query flooding and it performs well in networks with infrequent events. The drawback of RR is the high overhead for maintaining agent and event tables.

Although hybrid protocols could theoretically integrate favorable features of both proactive and reactive protocols, research on hybrid protocols has so far been limited. The main idea of hybrid protocols is achieving a balance between the energy-efficiency of on-demand route setup and the rapid data transmission in proactive routing. A key challenge for hybrid protocols is determining this optimal balance and providing flexible means for tuning the protocol for different applications.

4.2 Topology

In this section, we discuss the effect of topology on the performance of a routing protocol for ad hoc and sensor networks. Within this context, topology emerges here as a cross-layer aspect, since it impacts the routing behavior, as well as the MAC behavior. We examine routing protocols in light of the four topology categories we introduced in Section 3.4. We further identify and present the multi-hierarchy topology in our discussion for routing protocols.

4.2.1 Single Hop and Centralized Topologies

Routing is straightforward in both single hop and centralized topologies. In a single hop topology, each node can broadcast its packet to all neighbors. The intended receiver hears the packet and processes it. All other nodes discard the packet after examining the destination address.

In a centralized topology, the central base station is within range of all nodes. When the base station receives a transmitted packet, it simply determines where it needs to forward the packet in order to reach the intended receiver. The next hop could be either the destination itself, or an intermediate node. In either case, all nodes other than the base station are not involved in routing decisions.

Although single hop and flat topology networks involve simple routing mechanisms, they lack the scalability for supporting large scale ad hoc and sensor networks.

4.2.2 Multiple Hop Flat Topology

A multiple hop flat topology assumes all nodes are homogeneous in resources and that network traffic is random. In most cases, a multiple hop flat topology can help promote fairness and load balancing in the network.

Many general purpose ad hoc network routing protocols adopt a multihop flat topology [66, 71, 72, 76, 182]. Flexibility is a primary benefit of using this topology. Because the protocols view all nodes equally, multiple hop flat topology protocols can easily handle a failure or congestion at a particular node by rerouting data along new paths, provided there is sufficient node density to do so.

Some sensor network-specific routing protocols also adopt a multiple hop flat topology. For example, SPIN adopts a multiple hop flat topology which assumes that all the nodes are possible data sinks. All the nodes in SPIN are also capable of acting as sources by issuing ADV messages. Another sensor network routing protocol that adopts this topology is Minimum Cost Forwarding (MCF) [80]. MCF enables nodes to independently compute their routing cost from a data sink. Discovery of routing cost occurs progressively by summing up the cost of the immediate link with the next hop and the reported cost of the next hop to the sink.

In larger networks with a multihop flat topology, the hop count to certain destinations and the routing state stored at each node could become too large for the routing protocol to handle. Furthermore, as the average number of route hops increases in a network, so does the potential for transmission errors and retransmissions. These shortcomings of multihop flat topology networks have led to the design of protocols with a clustered topology to support large scale networks.

4.2.3 Clustered Topology

In a clustered topology, a subset of the nodes act as leaders or cluster heads. All the other nodes associate themselves with at least one cluster head. A node associated with a particular cluster head i is said to belong to cluster i. The cluster head acts as a virtual base station for the nodes within its cluster. Cluster head functionality may include forwarding the data of nodes within its cluster, managing its cluster's topology, and scheduling the transmission of nodes. A clustered topology reduces the energy consumption of most nodes by shifting most of the network management burden to the cluster head. Cluster head selection may be either static or dynamic. Topologies with a static or fixed cluster head are less fault tolerant because the cluster head represents a single point of failure for the nodes in the cluster.

Routing protocols that enable dynamic selection of the cluster head eliminate the single point of failure issue, but they involve more communication delay and control overhead for the repeated election of the cluster head. Cluster head Gateway Switch Routing (CGSR) [81] is a proactive clustered routing protocol that enables nodes to dynamically elect cluster heads. In CGSR, the cluster heads are responsible for forwarding node data, so nodes only need to store information on how to reach their cluster head. Communication among cluster heads in CGSR uses the DSDV protocol.

Cluster Based Routing Protocol (CBRP) [82] proposes a clustering approach to minimize on-demand route discovery traffic. Nodes in CBRP periodically exchange HELLO messages for maintaining neighbor tables that store cluster membership information for each neighbor. Nodes also maintain a 2-hop topology link state table. CBRP also proposes mechanisms that enable adjacent cluster discovery by including a cluster adjacency table within the HELLO messages. To send a packet, a sender node floods all known cluster heads with a RREQ message. RREP is sent back to the source along a reversed "loose source route" of cluster heads. In other words, the destination node sends RREP to its own cluster head, which in turn forwards the data to the cluster head in the sender's own cluster. Finally, Each cluster head along the loose source route incrementally computes a hop-by-hop strict source route.

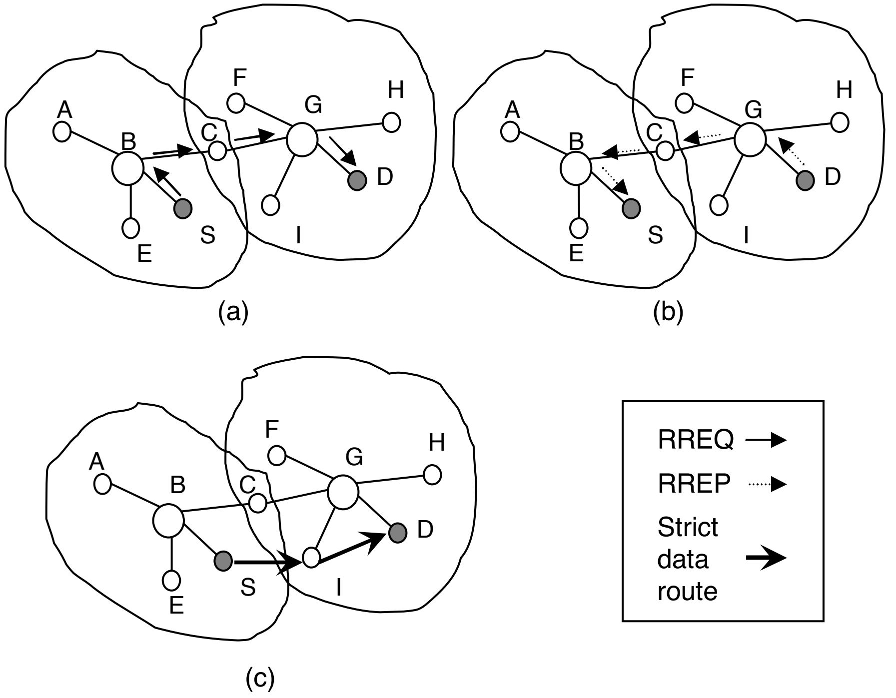


Fig. 4.7. CBRP route setup

Figure 4.7 illustrates the route setup process in CBRP. The sender node S wants to send data to the destination D. To find a route to node D, node S floods the accessible cluster heads B and G with RREQ messages (Fig. 4.7(a)). Node G determines that the RREQ is meant for a node within its cluster, so it forwards the RREQ to node D. In response, node D sends a RREP message to its cluster head node G, which forwards the RREP on the reverse route of RREQ back to S (Fig. 4.7(b)). The reverse path is a loose source route since it traverses the cluster heads and doesn't necessarily represent the shortest route between S and D. Upon receiving RREP, the cluster heads G and B compute the strict source route through which data will flow from S to D (Fig. 4.7(c)).

A class of sensor network routing protocols, including LEACH, PEGASIS, and TEEN, also use a dynamic cluster topology. The goal in all three protocols is to balance energy consumption among the sensor nodes, which are assumed to be homogeneous, by rotating the role of cluster head. LEACH segments time into rounds, where the initial setup phase of each round handle the cluster setup for the round. Nodes use a local algorithm to decide whether to be cluster heads for this round, and non-cluster nodes associate with one of the neighboring cluster heads. PEGASIS relies on geographic information that is locally available at each node in order to rotate the role of cluster fairly among the nodes.

Clustered topologies introduce a hierarchy into the network topology by focusing the routing at a few designated nodes. Generalizing the hierarchical model leads us to the topic of the next section: routing in multilevel hierarchical networks.

4.2.4 Multilevel Hierarchical Networks

Multilevel hierarchical (MLH) topology networks represent a superclass of clustered networks. Instead of designating nodes as simply cluster heads or data nodes, multilevel hierarchical protocols enable several hierarchical levels, forming a logical tree routing structure. Cluster heads at one level can form clusters of their own and select a higher level cluster head. MLH protocols provide more scalability over other topology types. A comprehensive survey of MLH routing protocols is available in [92].

Adaptive Routing using Cluster Hierarchies (ARCH) [93] is a MLH protocol that enables neighboring mobile nodes to exchange periodic HELLO messages for adapting the number of levels in the hierarchy to the network size. If the network size increases, nodes will form an additional hierarchical level and vice versa. ARCH provides route robustness by ensuring that non cluster head nodes are connected to more than one cluster head, so that if a cluster head becomes unavailable, a node uses the alternative cluster head.

Hierarchical State Routing (HSR) [83] is another MLH protocol that distinguishes between the physical routing hierarchy, which is dictated by geographic locations of nodes, and the logical topology of the network, determined by logical node groups. HSR borrows concepts from the Mobile Internet Protocol [84], such as home agents and tunneling, to implement a group mobility model that is applicable for application scenarios in which nodes move as a group, such as battle conditions or disaster recovery. The logical partitions and mobility model reduce the update overhead in highly mobile situations. The drawbacks of HSR are the possibility for non-optimal routing, which is incurred by tunneling, and the increased complexity.

In sum, MLH protocols adopt the most scalable and general topology. Large scale networks can exploit MLH topologies effectively, but small or medium scale network applications should consider the added complexity and delay of maintaining a multilevel hierarchical topology.

4.3 Multipath Routing

The traditional routing approach is to forward packets along the best single route from a source to destination. Although this approach is simple and straightforward, it suffers from poor fault tolerance. Any break in the single route from source to destination requires the determination and establishment of another suitable route. Route reestablishment involves increased communication overhead and delay. In addition, applications that require continuous data transmissions may not tolerate interruptions during route reestablishment.

More recent research has explored establishing multiple paths between a source and destination, which gives rise to the term “multipath routing”. Multipath routing not only reduces the fault tolerance of a routing protocol, but it also provides better load balancing in the network. Nodes can spread the traffic more evenly among several paths to avoid the creation of hot spots in the network. Finally, multipath routing also enables higher bandwidth for traffic streams by aggregating the bandwidth capability of several paths. Of course, multipath routing also has drawbacks such as increased complexity and overhead. Nodes have to store more state and exchange more control messages for maintaining multiple paths.

For example, Chen et al. [105] propose multiple disjoint paths with one primary path and multiple secondary routes which are maintained through periodic control signalling. Chen et al.’s protocol routes most of the packets along the primary path, and it uses the secondary paths when the primary path becomes congested or unavailable. The work in [105] uses end-to-end error control in which the final destination can notify the sender of errors along the path.

Similar multipath routing protocols [36, 106] propose using partially disjoint paths instead. Reference [106] proposes per-hop rather than end-to-end acknowledgements, to better adapt to the dynamic network environment and to detect fault more quickly.

Meshed MultiPath Routing [108] departs from the concept of primary routes and it enables equal use of the established routes. The rationale behind the equal use of routes is to promote load balancing and to avoid the exhaustion of node resources along the primary routes.

In sum, multipath routing is promising for large networks because it ensures transmission continuity and fault tolerance. Establishing and storing the state for multiple paths may be too costly, so gradient-based protocols, such as directed diffusion, may be a better option for multipath routing.

4.4 Power-awareness

The small form factor and mobility of nodes in ad hoc and sensor networks impose the use of small and portable batteries with limited energy resources. Because the nodes function as routers in addition to their data transmission function, energy efficient mechanisms are critical in ad hoc and sensor networks. As a result, many proposed routing protocols attempt to incorporate energy considerations from across the communication stack into routing decisions, representing another cross-layer optimization. The discussion in Chapters 7 through 9 further elaborates on using power-related information for cross-layer optimizations.

Minimum Total Transmission Power Routing (MTPR) [94] is a power-aware routing protocol for ad hoc networks. MTPR is an on-demand protocol that assigns *link* costs proportionally to the transmission power required to maintain an acceptable Signal to Noise Ratio (SNR) for the link. Nodes then select the minimum energy path through a shortest path algorithm based on link costs. However, the resulting path in MTPR is not necessarily the path with the minimum number of hops, which may result in higher delays and more power consumption at intermediate nodes. In an attempt to overcome this drawback of MTPR, Minimum Total Transceiving Power (MTTP) [95] adopts a link cost metric that includes both the transmission power as well as receiving power.

While protocols based on transceiver power reduce power consumption in the network, these protocols do not ensure a longer network lifetime. To this end, researchers have proposed protocols that incorporate the battery energy of *nodes* into routing decisions. For instance, Minimum Battery Cost Routing (MBCR) [96] provides a battery cost function that is inversely proportional to the remaining battery capacity at each node. The cost of a path in MBCR is the sum of costs of *nodes* along the path. When all nodes have comparable residual battery capacity, nodes using MBCR select shortest hop routes. Otherwise, they select minimum energy routes. The drawback of MBCR is that it may overuse intermediate nodes because the path cost only minimizes the aggregated cost instead of individual node battery costs.

Toh proposes Conditional Max-Min BCR (CMMBCR) [186] which combines power considerations for both the transmission power on *links* and battery capacity of *nodes*. CMMBCR uses battery capacity instead of a battery cost function as a routing metric. Routing decisions in CMMBCR only consider paths with nodes that have sufficient remaining battery capacity. Among the eligible paths, CMMBCR chooses the path with the minimum transmission power.

Shah and Rabaey [187] introduce network survivability as an improved routing metric, particularly for sensor networks. In other words, their contention is that a protocol should maintain connectivity in a network for the longest duration possible. Unlike previous protocols, they propose storing several suboptimal paths at each sender in addition to the optimal path. The basis for this proposal is that always using the lowest energy path is not beneficial for the long-term energy health of a sensor network. The transmission can then use the multiple paths through a probabilistic scheme in order to balance energy consumptions across the multiple paths and to ensure that nodes close to data sinks do not deplete their resources early. This protocol adopts a routing metric combining link transmission power and residual battery capacity at nodes.

Jurdak [100] proposes a cross-layer routing cost metric for sensor networks based on the radio, processor, and sensor activity. In Jurdak's approach, a node compares the power consumption of a neighbor relative to the average power consumption of its other neighbors through a weighted sum of duty cycles of the all components of a sensor node. This routing metric replaces the remaining battery capacity with the radio duty cycle because battery voltage readings in sensor nodes are unreliable [101] and their fluctuations may lead to inefficient or oscillating routes. Chapter 10 provides further details on this metric.

Another perspective that implicity minimizes radio power consumption is to minimize the expected total number of packet transmissions (including retransmissions) for delivering packets to their final destination, as proposed by De Couto et al. [99]. Their strategy highlights the energy inefficiency of minimum hop count routing in the presence of lossy wireless links. Minimum hop routes typically maximize the distance traveled at each hop, which reduces signal strength and maximizes the loss ratio of data packets and the corresponding acknowledgement packets, causing many retransmissions. Another drawback of traditional routing metrics is their lack of consideration of link asymmetries, such as unidirectional links. To address these issues, De Couto et al. propose the expected transmission count (ETX) routing metric, which is inversely proportional to the product of the forward and reverse packet delivery ratios. By considering both forward and reverse packet delivery ratios, ETX addresses the asymmetric link issue. The dependence of routes on delivery ratios also improves throughput and minimizes spectrum use. In practical terms, nodes that employ the ETX routing metric snoop on the neighborhood links to keep track of the forward and reverse delivery ratio on each link. The ETX routing metric is by definition a cross-layer routing metric, as it employs link layer information for determining behavior at the routing layer.

4.5 Geographical Routing

Geographic routing protocols assume that location information is available at the nodes through GPS or similar localization systems. Nodes can exploit available location information to route data in the geographic direction of the destination, yielding a more efficient routing strategy.

The main challenge of geographic routing protocols is the implementation and resource cost for providing location information. For example, having a GPS receiver provides each node with absolute location coordinates. However, providing nodes with location information is not without cost – to equip all nodes with GPS receivers incurs considerable monetary cost and causes higher energy consumption at the nodes. More efficient techniques for localization include signal strength [85] or time-of-flight measurements [86] along with local triangulation to provide nodes with relative location information. Hybrid localization methods combine the absolute positioning capability of GPS with affordable and energy-efficient localization techniques. For example, a network could include a few nodes with GPS receivers which provide absolute location information and act as location servers. Other nodes compute their location relative to the GPS-equipped nodes in order to eventually infer their own absolute location information.

Beyond the issue of collecting location information, geographic routing protocols exploit available location information through different strategies. Greedy Perimeter Stateless Routing (GPSR) [87] uses location information to forward packets to nodes in the general direction of the final destination. Whenever a greedy path does not exist (i.e., the only path which requires data forwarding to move temporarily farther away from the destination), GPSR recovers by forwarding in perimeter mode, in which a packet traverses successively closer faces of a planar subgraph of the full radio network connectivity graph. This process repeats until a node closer to the destination is reached, at which point greedy forwarding resumes.

Location-aided Routing (LAR) [88] is another proposed geographical routing protocol that uses location information to limit the search space for a desired route. LAR defines the concepts of expected zone and request zone. The expected zone is the estimated region of the destination's current location, based on the most recent update of the destination's position and speed. To set up an on-demand route, the sender can limit route request messages to the expected region. While the expected region estimates the location of the final destination, the concept of request region specifies the region within which an intermediate node forwards the route requests. By limiting the forwarding of route requests, LAR yields significant reductions in control overhead over flooding.

Grid Location Service (GLS) [89] is another geographic routing protocol that aims to provide the following features: load balancing; fault tolerance; local collaboration; and limited storage overhead. GLS employs the concept of location servers that store location information. Each node acts as a location server on behalf of some other nodes. The selection of location servers avoids election techniques. Instead, each node probabilistically determines a subset of nodes to act as its location servers in order to spread the load evenly among nodes. The location servers of a node are dense in regions close to the node, and they become more sparse as we move further away from the node. Nodes can query the location servers of the intended destination to determine the current location of the destination.

Seada et al. [90] propose new strategies for geographic routing under the realistic conditions of lossy wireless links in sensor networks. Their contention is that traditional geographic routing protocols employ a maximum distance greedy forwarding technique, without regard to the variance of quality in longer distance links. With a rationale similar to that of ETX, Seada et al.'s ultimate goal is the maximization of the number of packets delivered to the sink for each unit of energy spent in a geographic routing network. In fact, the collection of link variables is identical to the technique of De Couto et al. [99]. Seada et al. also investigate cross-layer routing metrics that rely on link layer information, most notably the PRR metric [91] that incorporates information on a link's SNR, encoding ratio, and frame length. For more selective geographic forwarding, they propose several techniques for blacklisting neighboring links in the route decision process, based on combinations of distance and reception rate considerations. Based on analysis, simulations and test-bed studies, they conclude that the most expressive routing metric for geographic forwarding is the product of PRR and distance.

In sum, geographical routing trades off reduced control overhead and reduced delay in forwarding data with the increased energy and communication overhead for collection and maintenance of accurate location information. The implementation of geographical routing through a hierarchical or clustered topology can limit routing state storage at each cluster or level, providing a high degree of scalability. Greedy geographic forwarding decisions should consider the energy-distance trade-off to maximize throughput and minimize packet losses in lossy wireless environments that are characteristic of ad hoc and sensor networks.

4.6 Quality-of-Service

Energy-efficient operation is a key goal for many applications in ad hoc and sensor networks, but there are also other quality of service metrics that may be even more important for particular applications. For example, an ad hoc network for disaster relief requires minimum delay in the network in order to maximize chances for finding survivors. In another application, a sensor network for detecting toxic leaks in a factory may prioritize high reliability over all other considerations. As a result, routing techniques for ad hoc and sensor networks have been developed for supporting more general QoS guarantees. Enforcing QoS policies usually benefits from cross-layer design since it may require access to information from higher layers, such as application meta data, and from lower layers, such as acknowledgements from the logical link control layer.

QoS-based routing protocols aim at satisfying one or more performance goals at the network or node level. Performance goals include robustness, low delay, high reliability, energy efficiency, and high throughput.

Protocols for general purpose ad hoc networks should provide QoS on the node level. Nodes in ad hoc networks operate as independent entities, and they have individual communication goals. Therefore, the QoS guarantees in ad hoc networks must enable per node and even per connection performance optimization.

Core Extraction Distributed Ad hoc Routing (CEDAR) [198] is a QoS routing algorithm with a primary goal to provide robustness. CEDAR first extracts a subset of nodes, the so-called core, through local computations to maintain network routing state. The core nodes propagate link state information to remote nodes in the network. In particular, core nodes report the stable high bandwidth links and the unfavorable links to remote nodes, which can then use this information for routing decisions. Non-core nodes only need to determine their routing path to the appropriate core node. Core nodes then forward the packets in the direction of the destination.

Baldi et al. [175] propose a generalized cost function for supporting admission control in wireless ad hoc networks that use Ultra Wide Band (UWB) radio. Their cost function is a weighted sum of several link cost metrics: transmission power; link setup; interference; quality; and delay. The weights of individual cost metrics can be customized according to different applications and hardware. For instance, a delay-sensitive application might assign a high weight for the delay cost metric. The cost function also enables the inclusion of additional cost metrics as needed. Admissible links or paths must satisfy cost constraints in the network. Although originally designed for UWB networks, this cost function can be adapted to more general ad hoc networks and can be integrated as the cost metric of many cost-based routing protocols.

Unlike most general purpose ad hoc networks, providing QoS guarantees in sensor networks must consider network-wide performance goals, since sensor networks typically have a single operator or user. For example, prolonging the lifetime of the network as a whole is more important than optimizing the lifetime of a single node in a sensor network.

Sequential Assignment Routing (SAR) [103] is a QoS routing protocol for sensor networks. SAR assumes that the sensor network has a single data sink. Rather than generating a single path to the sink, the protocol enables each node to generate multiple paths to avoid route recomputation overhead whenever a route fails. SAR builds multiple trees, with each tree rooted at a one-hop neighbor of the sink, in order to provide multiple paths to the sink. The protocol also adopts a general QoS metric representation. Each data node can choose its path to the data sink based on its stored information regarding the energy resources and the cumulative quality-of-service parameters associated with each path towards the sink.

Energy Aware QoS Routing [104] is a protocol for sensor networks that aims at providing bandwidth and delay guarantees in an energy-efficient manner. The protocol addresses networks with both real-time and non real-time traffic. It attempts to minimize end-to-end delay for real-time traffic while maximizing throughput for non real-time traffic. The protocol computes cost for links independently, except for end-to-end delay which is aggregated for links along a path. The protocol's cost function incorporates transceiver power, sensing power, residual battery energy, energy consumption rate, and error rate. It also includes a metric that quantifies the cost of switching a node from inactive to forwarding state, and another metric that penalizes nodes with a large forwarding load.

In sum, reliability, delay, robustness, and throughput are important performance considerations for ad hoc and sensor networks. Routing protocols that support QoS enforcement tend to provide tunable mechanisms that enable network designers to determine the relevant QoS parameters and their relative importance.

5

Transport and Middleware Layers

This chapter focuses on current ad hoc and sensor network mechanisms at the transport, session, and presentation layers in the OSI network model, shown in Figure 5.1. Some of the current ad hoc and sensor networks efforts do not require the services and functionality of these higher layers, so they build applications directly on top of the network layer. However, the rapid maturing of the ad hoc and sensor network applications will soon require the more sophisticated services of the transport, session, and presentation layers, such as flow control or session management.

This chapter is structured as follows. Section 5.1 examines transport protocols for ad hoc and sensor networks and their relation to traditional transport protocols. Section 5.2 surveys existing middleware approaches, which bundle up the functionalities of the session and presentation layers of the OSI model, providing generic interfaces between applications and underlying network implementations.

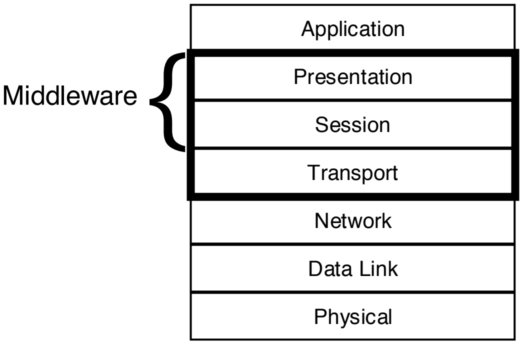


Fig. 5.1. The higher networking layers in the OSI reference model

5.1 Transport Layer

The transport layer is responsible for flow control, congestion control, end-to-end connection setup, and end-to-end reliability. The most popular transport protocols for Internet applications are Transmission Control Protocol (TCP) [109] and the User Datagram Protocol (UDP) [110]. TCP is a reliable byte-stream connection-oriented protocol, while UDP is a best-effort connectionless protocol. Both TCP and UDP were originally designed for wired networks long before the conception of ad hoc networks. As such, these protocols do perform well in the highly dynamic wireless environment of ad hoc and sensor network applications. This has led to several enhancements over the traditional protocols and to the proposal of new transport protocols for ad hoc and sensor networks, which we discuss in this section.

The remainder of this section is organized as follows. Section 5.1.1 provides a brief description of the traditional transport protocols and why they are incompatible with ad hoc and sensor networks. The remaining two subsections present transport protocols specifically designed for ad hoc and sensor networks, most of which protocols integrate cross-layer optimizations. Section 5.1.2 discusses the general-purpose transport protocols for ad hoc networks that focus on preserving session connectivity and information flow in highly dynamic environments. Section 5.1.3 explores novel sensor network transport protocols, which tend to exploit application specific information for improving performance.

5.1.1 TCP and UDP

UDP provides a transaction-oriented datagram service above the IP layer without providing any guarantees of delivering or maintaining the order of packets. The main contribution of UDP is to support several concurrent applications over the same IP interface through the introduction of ports that are bound to the applications.

The most widely used transport protocol, TCP, currently accounts for more than 90% of Internet traffic. TCP's design goals include:

1. **Reliability**: TCP provides reliable end-to-end data delivery through ACK messages.
2. **Congestion control**: TCP attempts to limit excess traffic in the network in order to maintain acceptable transfer rates.
3. **Flow control**: a process by which a receiver adapts the per-flow traffic to its reception capacity.

In order to perform both congestion control and flow control, TCP uses an elastic congestion window. The size of the congestion window is adjusted according to the traffic requirements in the network.

The basic algorithm in TCP works as follows. The TCP sender initially sets the congestion window size to 1 packet, proceeds to send its first data packet, and sets a local timer. If the sender receives an ACK message before the timer expires, then the sender doubles the congestion window to 2 packets, sends the next 2 data packets to the destination and sets a new timer. As long is all the ACK messages are received in time, the sender doubles the window again. This process is referred to as "slow-start" of TCP. The TCP receiver also includes in its ACK messages the current buffer capacity at the receiver, referred to as the receiver window. The slow-start phase continues until the window size reaches a preset slow-start threshold, at which point the sender begins increasing the window size by 1 packet upon the reception of every successful ACK. This phase is called congestion avoidance, as it tries to anticipate that the exponential growth in window size during the slow start phase will surely cause congestion. Congestion avoidance persists until the sender's window size reaches the receiver window size and remains constant after that. Whenever an ACK message does not arrive in time at the TCP sender, the sender assumes that the packet loss is due to congestion and it invoke TCP's congestion control mechanisms. Subsequently, the sender resets its congestion window to 1 packet, resets its slow-start phase, and halves the slow-start threshold. The sender then proceeds in doubling the congestion window at every successful ACK and the process continues as described above.

The congestion control mechanism in classic TCP causes oscillations in the congestion window since every lost or duplicate ACK drops the congestion window back to 1 packet and resets the slow-start phase. Several extensions have been proposed [111–113] for improving the performance of TCP over traditional wired networks. One prominent extension is TCP Reno [111] in which the congestion window is set to half of its current value upon a failure to receive ACK or upon reception of a duplicate ACK. Upon reception of additional duplicate ACK messages, TCP Reno increments the congestion window linearly at 1 packet per duplicate ACK reception.

Because their design targets traditional networks, the congestion control, flow control, and end-to-end acknowledgement mechanisms yield poor performance when TCP is deployed in an ad hoc or sensor network. To begin with, the TCP congestion control mechanism always assumes that a lost ACK packet results from network congestion. In ad hoc and sensor networks, the distributed nature of the network leads to more frequently missed ACKs due to MAC layer issues (such as collisions, hidden terminal problems, and co-channel interference) and routing issues (such as node mobility or disappearance, intermittent or unidirectional wireless links, and path breaks). The high possibility of missed ACKs causes the TCP congestion window to drop back to 1 packet more frequently, yielding low throughput and low channel utilization.

MAC layer issues can cause degraded performance for TCP over ad hoc and sensor networks. Collisions and hidden terminal problems are more likely in ad hoc and sensor network links because of the distributed channel access mechanisms. Furthermore, the broadcast nature of the wireless channel causes interference among coexisting transmissions. These MAC layer issues lead individual links to suffer higher packet losses along a TCP path, resulting in a higher number of missed ACK packets at the receiver.

Routing issues also contribute to the degradation of the congestion window mechanism of TCP over ad hoc and sensor networks. Because of the dynamic topology and node mobility, ad hoc and sensor network routes are longer and less stable than traditional networks, which leads to more frequent lost ACKs. Also, unidirectional links cause ACK losses since the end-to-end reliability mechanism in TCP assumes bidirectional links. Finally, the multi-path routing protocols in Section 4.3 that distribute data delivery over multiple routes may cause packets to arrive out-of-order at the receiver. Out-of-order packets cause the receiver to issue duplicate ACKs, leading the sender to invoke the congestion control mechanism.

5.1.2 Ad Hoc Network Transport Protocols

To overcome the drawbacks of traditional transport protocols, researchers have proposed several transport protocols that are customized for ad hoc networks. Some of the proposed protocols are modifications of the original TCP, while others are completely new protocols.

TCP Enhancements

The dependence of TCP performance on lower layer attributes has led to the design of TCP-based enhancements that exploit cross-layer information for better operation over ad hoc and sensor networks. All of these enhancements address the high probability of packet losses due to the unreliability of individual links and routes. In particular, the common direction is to provide mechanisms to notify the sender of the cause of packet losses and to freeze the congestion window algorithm while the problem is rectified.

One of the enhancements for ad hoc and sensor networks is TCP with explicit link failure notification (TCP-ELFN) [114]. TCP-ELFN enables an intermediate node to detect a link failure along an active path and it notifies the TCP sender through control messages to freeze the retransmission timers and the congestion window. The TCP sender then uses periodic probe packets to determine when the receiver is reachable, at which point it restarts the timers and data transmission. Because TCP-ELFN relies on link failure notification, it has little dependence on routing layer functionality. One drawback of TCP-ELFN is that a new route to the receiver may not support the congestion window of the old broken route.

Other enhancements use similar feedback mechanisms [115–117], but these enhancements notify the sender of a break in the path rather than an explicit specification of which link has failed. Feedback based TCP (TCP-F) [115] enables the node that detects a path break to notify the sender through routing layer control messages. Upon notification of a path break, the sender enters into snooze state, freezing the data transmission, congestion window size, and all timers until a path to the destination can be reestablished. The node that detects the path break also propagates this knowledge to other nodes. Any intermediate node on the original path that has an alternate route to the destination can notify the sender of this route, at which point data transmission resumes. The advantage of TCP-F lies in its simplicity and effectiveness. One of its drawbacks is the requirement for network layer support for path break notification. Another drawback is that the new snooze state for TCP-F requires modifications with original TCP state machine, which makes TCP-F a less attractive option for widespread adoption.

To remedy this issue, Ad Hoc TCP (ATCP) [117] uses a similar feedback mechanism to TCP-F without creating a new state in the TCP state machine. Instead, it introduces a new ATCP layer between the transport and network layer. ATCP distinguishes between three causes of packet loss: network partition; transmission errors; and congestion. In the case of a network partition, ATCP puts the sender in persist mode so that it doesn't needlessly retransmit packets. For transmission errors, the sender retransmits packets without invoking the congestion control. Finally, in case of congestion, ATCP invokes the congestion control mechanism at the sender. Because ATCP mechanisms work at a separate layer that is transparent to the transport layer, ATCP enables full compatibility with traditional TCP. However, the new ATCP layer requires modification of the interfaces that connect the network and transport layer. Also, ATCP still requires network layer support for notifying the sender of a path break.

New Transport Protocols

The advantage of TCP-based enhancements is that they offer some degree of interoperability with the existing TCP systems. However, new protocols can offer better optimization capability for stand-alone ad hoc networks that do not require a connection with the Internet. New transport protocols tend to exploit cross-layer information sharing and notifications to better guide congestion control algorithms.

For example, Liu et al. have proposed the application controlled transport protocol (ACTP) [118], a lightweight transport protocol for ad hoc networks that relies on cross-layer information sharing. The key idea of ACTP is to equip other layers in the stack with ample information to perform reliable packet delivery. For example, ACTP provides the network layer with packet priority information but it leaves it to the network layer to enforce the packet priorities. Similarly, ACTP informs the application layer of packet delivery status, but it assumes that the application layer enforces reliability mechanisms. Thus, ACTP provides a large degree of freedom in reliability and priority implementation. However, its lack of compatibility with TCP may limit its widespread adoption. Additionally, ACTP lacks a congestion control mechanism which may quickly lead to network saturation in larger networks.

Ad hoc Transport Protocol (ATP) [121] is another novel transport protocol for ad hoc networks. ATP's design relies on the use of cross-layer information sharing, particularly by using lower layer information at the transport layer. It also employs explicit feedback from other network nodes to assist in the transport layer mechanisms. The feedback from other nodes includes initial rate feedback for startup rate estimation, progressive rate feedback during congestion, and path failure notification. Intermediate nodes in ATP maintain state information on queueing and delay, aggregated for all packets traversing the node. The intermediate nodes do not have to maintain any per-flow state information, which improves the scalability of ATP. The receiver also periodically provides the sender with reliability and flow control feedback by collating the rate feedback from intermediate nodes. ATP exhibits improved performance and scalability, in addition to the increase in the stability of the congestion window. ATP's drawback is its lack of compatibility with the widely deployed TCP base.

Fu et al. [119] developed the Ad Hoc TCP Friendly Rate Control (ADTFRC) transport protocol specifically for multimedia applications in ad hoc networks, based on the earlier TFRC algorithm for wired networks. As in ATCP, ADTFRC addresses the problem of false identification of congestion in ad hoc networks due to other frequent events that cause packet loss in these networks. It uses an end-to-end paradigm to enable the receiver to differentiate between different network events, such as route breaks or congestion, based on independent measurements of multiple metrics, including interpacket delay difference and the short-term throughput. The cross-validation of multiple metrics enables the receiver to distinguish congestion events that require congestion control from other network events. The benefit of ADTFRC is that it reduces the probability of misinterpreted network events. However, it does not provide fully robust identification of network states. ADCTP's design also aims at providing TCP-friendly behavior, to ensure fairness when ADCTP coexists with TCP in the same network. Despite its TCP-friendly behavior, ADCTP's introduction of new network states does not ensure backward compatibility with traditional TCP.

5.1.3 Sensor Network Transport Protocols

There is an ongoing debate whether there is a need for reliable transport mechanisms in sensor networks. Many sensor network applications, such as environmental monitoring or location systems, do not require data delivery guarantees as they can tolerate occasional data loss. However, other sensor network application classes, such as industrial monitoring or medical monitoring require delivery guarantees as they are more sensitive to packet loss. Another application that requires reliability is network reprogramming, such as Deluge [120]. Instead of manually retasking the sensors, the network operator can inject new parameter settings, upgraded programs, or new programs into the network. The new programs then propagate among the nodes in an epidemic manner until all the nodes have received the new program. Uploading program code to sensor nodes requires strict guarantees of data delivery, since the loss of even 1 packet may render the program useless.

Several researchers have proposed sensor network transport protocols to serve the needs of applications that require reliability. Wan et al. have developed the Pump Slowly Fetch Quickly (PSFQ) [122] transport protocol for sensor networks that maintains packet order. The idea of PSFQ is to keep packets that are transported along a multihop path in their original order. Figure 5.2 illustrates the operation of PSFQ through an example [123]. If an intermediate node *B* that expects to receive a packet with sequence number 2 receives a packet with a sequence number 3, then node *B* sends a request for retransmission of the missed packets to the previous hop on the path. Node *B* also stores all incoming packets with a sequence number larger than 2 and refrains from forwarding these packets along the path until it receives the packet with sequence number 2. When packet 2 arrives successfully at node *B*, normal forwarding resumes. PSFQ avoids the delay of end-to-end retransmissions through the use the hop-by-hop retransmission of PSFQ to maintain the original packet order.

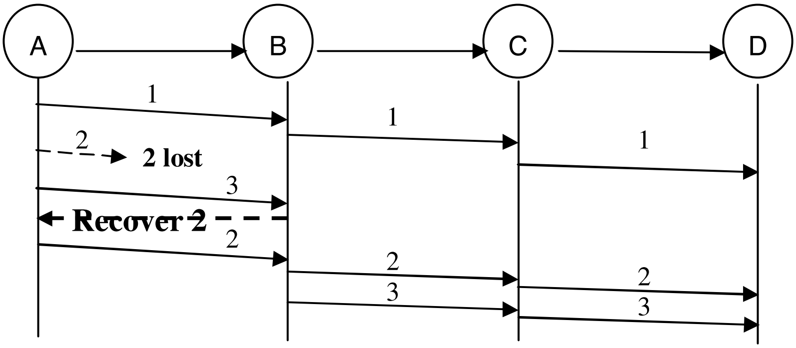


Fig. 5.2. PSFQ example

A related work in [124] introduces the Event to Sink Reliable Transport (ESRT) protocol for sensor networks. ESRT is specifically designed for typical sensor network applications in which the sensor nodes report events to a single data sink. In addition to providing congestion control, ESRT's goals include:

1. **Self-configuration**: The self-configuration feature of ESRT operates mainly at the sink node, which monitors incoming network traffic to detect signs of network congestion.
2. **Energy-awareness**: The sink can dynamically adapt the data reporting frequency in the network to maintain the desired degree of reliability. If

the sink detects a higher reliability rate than required, then the sink can reduce the reporting frequency to conserve energy.

3. **Collective Identification**: In contrast with traditional end-to-end approaches, ESRT adopts a collective identification approach. ESRT's design builds on the fact that the sink node only needs to collect data about events, and it does not need to know the individual identities of nodes. Thus, end-to-end connections are not supported in ESRT.

Most of ESRT's decision making functionality is located at the data sink, which has more processing and energy resources than typical nodes. ESRT always converges to the optimal operating region that balances reliability and reporting frequency. Another strength of ESRT is its custom design for sensor networks, which promotes energy efficiency, adaptivity, and reliability. One drawback of this protocol is the variation of reporting frequency with reliability, which may be unsuitable for certain applications that have strict requirements on data reporting times. Another disadvantage is the assumption of a single sink, which creates a single point of failure and prevents application of ESRT to more general topologies.

The Sensor Transmission Control Protocol (STCP) [125] has many common goals with ESRT: controlled variable reliability; base station controlled network; and congestion control. SCTP also considers the energy-reliability tradeoffs. The main difference between STCP and ESRT is that STCP adopts a session-based end-to-end approach like TCP. STCP adopts two different mechanisms for supporting reliability for continuous and event-driven traffic. For continuous traffic, SCTP determines if packets were lost based on the expected arrival time, which is computed from the data reporting period. This mechanism may prove difficult to implement in practice because delay between two data update periods may vary considerably due to route changes. For event-driven traffic, the base station provides positive acknowledgements to the sender. For both continuous and event-driven traffic, the base station notifies the sensor nodes of timeouts through negative acknowledgement messages. SCTP also acknowledges that data centric application may require data aggregation to limit data volume. In this case, the base station does not generate ACK and NACK messages in order to avoid control overhead. STCP's adoption of flow-based traffic and individual node ID's supports more granular data reporting than ESRT. However, its connection-oriented nature may create too much control overhead in the network. Another drawback is STCP's unrealistic reliance on timely data reporting from the sensor for determining timeouts.

5.2 Middleware

The middleware layer typically provides applications with high level abstractions of the underlying communication system, enabling the application to tune network behavior with minimal information on underlying platforms. In most cases, the middleware layer provides services typically from the OSI session and presentation layers.

Middleware models tend to provide the middleware layer with visibility to network aspects from all the lower layers, constituting an early and implicit attempt at cross-layer design. Chapters 7 and 8 present more recent cross-layer design approaches that integrate the interaction across layers into the system architecture. In this section, we overview the functionality of middleware in the context of ad hoc and sensor networks in order to expose potential cross-layer optimizations involving the middleware layer. We focus here on the interactions and network management aspects of the middleware, so the software paradigms that characterize the middleware systems are out of the scope of this discussion.

The middleware layer lies between the operating system and the application. Traditional network models run with well-established operating systems, so middleware for these models has clear cut functionality and interfaces. In contrast, ad hoc and sensor networks are developing research fields for which operating systems have not reached the same maturity and stability as traditional networks. As a result, the design of middleware for ad hoc and sensor networks is still in its infancy.

The remainder of this section visits the design issues in ad hoc and sensor network middleware that relate to cross-layer design. Section 5.2.1 explores the middleware design issues for ad hoc networks, while Sect. 5.2.2 focuses on sensor network middleware issues.

5.2.1 Middleware for Ad Hoc Networks

Our interest in middleware stems from its similarity to cross-layer design in sharing information across layers. The middleware's role is to observe the state of lower layers in order to adapt network behavior to the application requirements. The next part of the book reveals that cross-layer design attempts to achieve similar goals through a more holistic approach that adopts cross-layering within the network architecture.

Middleware design for ad hoc networks has attempted to build on the extensive middleware concepts for traditional networks. However, several additional design challenges characterize the development of middleware for ad hoc networks [126, 128–130]:

- **Heterogeneous node resources**: Ad hoc and sensor networks may include nodes of varying resources. For example, in an ad hoc network of laptop computers and hand-held computers, the former have faster CPUs, more memory, and larger batteries. The middleware should be aware of these resource differences to better distribute the network load.
- **Nature of connections**: Wireless connections in ad hoc and sensor networks exhibit lower bandwidth and higher error rates due to the more

variable wireless channel. Node mobility also leads to intermittent connections, adding another level of complexity. The role of middleware is to insulate applications from these lower layer details.
- **Network partitioning**: Dynamic node membership in ad hoc networks may cause the network to split into several partitions that cannot communicate. Middleware design must employ mechanisms such as data replication and asynchronous communication to maintain smooth operation of the application in anticipation of renewed network connectivity.

For example, Epidemic Messaging Middleware for Ad Hoc Networks (EMMA) [132] is an example of middleware targeting ad hoc networks. EMMA builds on the concept of message-oriented middleware (MOM) for traditional systems. The attractive feature of MOM is the delay tolerant asynchronous traffic model, which suits the intermittent connections in highly dynamic ad hoc networks. EMMA takes a step in the direction of cross-layering by enabling the middleware layer to collect information from both the software and hardware components of a node to adapt its behavior. In fact, EMMA disregards traditional layer boundaries by considering the middleware and network layers jointly as the *communication layer*.

5.2.2 Middleware for Sensor Networks

The diversity of the sensor network application space has motivated the development of middleware to enable various application to operate on top of sensor network platforms. In addition to the challenges of ad hoc network middleware, middleware design for sensor networks involves the following additional challenges [129, 133]:

- **Resource limitations:** Sensor nodes typically have fewer memory, processing, bandwidth, and energy resources compared to general ad hoc network applications. Middleware components have to be lightweight to fit the tight constraints of the sensor nodes.
- **Cooperative applications:** Nodes in sensor networks usually cooperate to achieve a common application goal. Sensor network middleware must enable nodes to efficiently share their available resources and to perform in-network processing when required.
- **Scale of deployments:** Sensor networks are expected to have hundreds to thousands of nodes, which is at least an order of magnitude higher than traditional networks and general ad hoc networks. This suggests the need for scalable middleware that is capable of supporting such large deployments without performance degradation.
- **Adaptive fidelity:** One of the main tasks of the middleware layer is to map application performance requirements to protocol parameters. The highly dynamic state of sensor nodes may require applications to tradeoff one performance metric for another. The middleware must adapt quickly

> enough to provide an adaptive bridge between the dynamic network environment and the application.

As an example of sensor network middleware, consider Middleware Linking Applications and Networks (MILAN) [129], which attempts to balance QoS requirements and energy efficiency of sensor networks by optimizing the sensor and network configuration, based on descriptions of application requirements. The application descriptions in MILAN consist of graphs that incorporate state-based changes in application needs. While traditional middleware sits between the application and the operating system, MILAN considers an architecture in which the middleware extends into the network protocol stack, to provide support for multiple network platforms. This feature of MILAN touches upon cross-layer design by penetrating the traditional layer boundary of middleware to enable more profound network optimizations.

6

Application Layer

Real-world applications drive the interest in ad hoc and sensor networks. Ad hoc and sensor networks are expected to solve practical problems or to enhance existing life activities. A wide range of applications motivates ad hoc and sensor network development efforts. The unique context of each application scenario requires highly customized performance guarantees. This chapter examines the proposed applications and the related performance requirements that characterize these applications.

The chapter is structured as follows. Section 6.1 focuses on ad hoc network applications, the categories into which they fall, and the application performance metrics. Section 6.2 examines the sensor network application classes, and their respective performance requirements.

6.1 Ad Hoc Networks

The scope of ad hoc network applications is quite broad. Ad hoc network applications include highly mobile intervehicular networks, delay-sensitive search and rescue networks, and highly reliable emergency response networks. In this section, we first provide an overview of existing application classes of ad hoc networks, along with concrete examples of each application class in Section 6.1.1. We subsequently explore the ad hoc network application performance metrics and their relative importance for the relevant application classes in Section 6.1.2.

6.1.1 Ad Hoc Network Application Classes

This section groups ad hoc networks according to their application domain, including [131] military, emergency services, and home/office applications. The goal of classifying networks based on their application domain is to unveil the distinct application characteristics of ad hoc networks and how they relate to lower layer behavior. As ad hoc networking is still an emerging field, we expect the scope of ad hoc network applications to keep on rapidly expanding.

Military Applications

Military applications have motivated early research on ad hoc networks. The ability to quickly set up a network among military units in hostile territory without any infrastructure support can provide friendly forces with a considerable tactical advantage on the battlefield. For instance, each soldier can carry a mobile device that represents one of the mobile nodes in an ad hoc network linking all soldiers, tanks, and other vehicles. Recent advances in robotics have also motivated the idea of automated battlefields in which unmanned fighting vehicles are sent into battle. Supporting military applications requires self-organizing mechanisms that provide robust and reliable communication in dynamic battle situations.

Emergency Services

Another promising application area for ad hoc networks is emergency services, including search and rescue and disaster recovery operations. As an example of search and rescue, consider an airline that attaches small wireless devices to the life jackets under each seat. Suppose that the plane has mechanical problems and has to make an emergency landing in the water. Once search and rescue teams arrive at the landing site, they are provided with detailed information about the location (the coordinates and potentially the depth) of the victims through the transponders. As a result, the rescue teams can more effectively locate and reach the victims. The mobile devices could also monitor the vital signs of victims, such as heart rate or breathing rate, to prioritize the rescue of victims that are still alive.

A similar application arises when disasters, such as earthquakes, blackouts, or bombings occur. The disaster may destroy existing communication infrastructure, preventing critical contact among emergency workers. The emergency response teams can set up ad hoc networks quickly to replace the destroyed infrastructure, enabling the teams to better coordinate their efforts.

Home, Office, and Educational Applications

Ad hoc networks also have applications in home and office environments. The simplest and most direct application of ad hoc networks in both homes and offices is the networking of laptops, PDAs and other WLAN-enabled devices in the absence of a wireless base station. Another home application that falls within the Personal Area Network (PAN) class is wire replacement through wireless links, as in Bluetooth. All periphery devices can connect to a computer through wireless Bluetooth links, eliminating the need for wired connections.

Ad hoc networks can also enable streaming of video and audio among wireless nodes in the absence of any base station. For instance, UWB provides a sufficiently high bandwidth (in the order of Gb/s) to support several multimedia streams. UWB-equipped nodes can autonomously set up an ad hoc network to stream high quality video and audio between several computers through wireless UWB connections.

Educational and recreational activities can also benefit from ad hoc networks. For example, students attending a classroom can use their laptops to obtain the latest class material from a professor's laptop as the class progresses. On the recreational side, the mobility and nomadic nature of ad hoc networks enables richer multi-user games that can incorporate user mobility and proximity into the virtual game environment.

6.1.2 Application Performance Metrics

This section has so far explored potential ad hoc network application scenarios. We now examine the typical ad hoc network application performance metrics, and their relation to the application scenarios. Most of the application performance metrics are highly dependent on parameters and mechanisms at lower layers in the communication stack, revealing potential for cross-layer optimizations.

Prior to presenting the application performance metrics, we define the following traffic types for ad hoc networks:

- Continuous traffic: refers to continuous data flow along a link or path
- Synchronous traffic: refers to time-sensitive data flow
- Bursty traffic: refers to generic application data without any specific temporal pattern

Reliability

Ad hoc network applications that cannot afford any data loss emphasize reliability as a main performance requirement. Military applications and emergency management services both fall into this category, since the loss of any data in these applications may endanger human life. As the previous chapters reveal, many lower layer network mechanisms provide reliable data delivery, such as an Acknowledgement message at the MAC or the transport layers. However, applications that run over unreliable communication stacks may need to define their own application-level reliability mechanisms, such as application layer acknowledgement mechanisms.

Throughput

Ad hoc network applications with continuous traffic, such as video streaming or multi-user gaming, require a guaranteed level of throughput. Applications typically only specify their throughput requirements with the assumption that lower layer mechanisms are capable of satisfying these requirements. Providing throughput guarantees in ad hoc networks is challenging because of node mobility, highly variant channels, and the connectionless nature of most ad hoc network communications. As Section 5.2 describes, replicating the data at several nodes can minimize the effect of node mobility on end-to-end application throughput. However, data replication may not work if node velocities exceed a certain level. Another useful concept that promotes throughput guarantees for ad hoc network applications is the reservation of bandwidth along virtual links or paths, which is also subject to mobility considerations.

Delay

Synchronous traffic applications also require delay related guarantees. A streaming application, such as video or audio streaming, may require limits on the end-to-end delay or the interframe delay variance. Military networks and disaster relief networks also place strict constraints on end-to-end delay since tardy data in these critical and highly dynamic situations may become useless. In contrast, standard data communications applications such as internet browsing or ftp may tolerate delay.

Provision of delay guarantees in ad hoc networks is challenging because of their highly dynamic nature. Implementation of delay mechanisms at the application layer is not desirable, since end-to-end delay is highly dependent on lower layer activities such as medium access (collisions and backoffs), routing (queueing and forwarding), and transport (congestion and flow control). Most current ad hoc networks exploit algorithms at lower layers, such as priorities, scheduling, and reservations, in an attempt to deliver time-sensitive data within the delay bounds.

Energy Efficiency

The mobility of battery-powered nodes in ad hoc networks may prevent frequent replenishment of the limited energy resources at each node. The limited node resources motivate the design of energy efficient mechanisms for ad hoc networks at every layer in the network stack. Although applications do not directly manage energy-related issues, they can specify energy bounds or requirements. For example, applications can specify minimum network lifetime requirements.

Energy efficiency is certainly a cross-layer aspect that can benefit all classes of ad hoc network applications. For example, tuning of modulation rates at the physical layer, adaptation of duty cycles at the MAC layer, and choosing routes that maximize network lifetime at the network layer can all yield energy savings. However, the inherent tradeoff between energy efficiency, delay, and throughput in ad hoc networks may limit the potential energy optimizations.

Applications with bursty traffic, such as home and office ad hoc networks, may be able to tolerate considerable delay in order to preserve valuable node energy resources. Other applications with continuous or synchronous traffic that prioritize reliability or delay provide the potential for only limited energy optimization.

Data Accessability

Timely access to data is crucial for supporting time-sensitive applications with synchronous traffic. Because of dynamic node membership and node mobility, providing applications with continuous access to data remains an open research issue. A potential solution to the data accessability problem is the selective replication of data across several nodes to ensure that at least one node can service any requests for the data. Delay bounds in synchronous traffic applications, for example in multimedia applications, determine the optimal degree of data replication. If the delay bounds are too small, then the overhead of data replication may overweigh its benefits.

Fairness

Fair access to the network among multiple applications that run concurrently is yet another performance goal for ad hoc network applications. Reservations of communication resources by high-priority traffic, such as synchronous or continuous traffic, may cause bandwidth starvation of applications with lower priority traffic. Applications should typically specify the extent of fairness to be enforced by lower layers. For example, the network can allocate 60% of the available bandwidth to streaming traffic and 40% to bursty traffic. The network can further specify the allowable bandwidth allocations among streams of the same traffic class, to ensure fair access to communication resources. Naturally, enforcing fair access in ad hoc networks adds another level of design and implementation complexity for the network mechanisms.

Multicast Support

The channel in ad hoc networks is inherently a broadcast medium, so all nodes within the vicinity of a transmission receive the signal at the physical layer. However, broadcasting at higher layers is not energy-efficient since all the unintended receivers of a signal waste energy in processing the packet before discarding it. At the other extreme, nodes may address specific neighbors through unicast transmission. Unicast transmissions currently dominate research for ad hoc networks. In between broadcasting and unicasting, multicasting provides an alternative in which a node targets a designated group of nodes to process the received signal. For instance, gaming and enterprise applications of ad hoc networks can exploit multicast groups for richer interactions among users.

The network layer is traditionally responsible for implementing multi-cast mechanisms. However, several works have explored application layer overlays for forming multi-cast groups in ad hoc networks [134, 135]. The attractive aspect of application layer multicast support is that it avoids changes to the network layer and it does not involve increased state maintenance at intermediate nodes. The drawbacks of application layer multicasting are the possibility of redundant packets at intermediate nodes in the mulicast tree, and the use of sub-optimal routes.

6.2 Sensor Networks

This section first considers the broad sensor network application classes based on their data dissemination strategies, providing examples for each class in Sect. 6.2.1. The discussion in Sect. 6.2.2 then focuses on the application performance layer requirements for sensor networks.

6.2.1 Data Dissemination

Sensor networks collect data from their physical environments, process the data, and eventually report it back to the network user. We refer to this set of actions as data acquisition and dissemination. Three broad classes of sensor network applications emerge depending on the factors that drive data acquisition and dissemination: time-driven; event-driven; or demand-driven.

Time-driven Sensor Networks

We motivate the discussion on time-driven sensor networks with an example application. Consider an underwater network of sensors that monitors physical indicators, such as temperature, salinity, pressure, and wave direction, to determine the level of pollution in the water. The network consists of several subsurface nodes that are deployed to sense the underwater environment. One of the nodes, referred to as base station or data sink, lies on the surface within communication range of the underwater nodes. The base station is also a gateway node that can communicate with the underwater nodes and a relay station at the shore. Because changes in underwater pollution levels can vary within a time scale of minutes, the underwater nodes sample their sensors and send the data via the base station every few minutes. Monitoring the underwater environment at fixed time intervals provides environmental experts that will analyze the data with a well-defined data granularity for predicting pollution in the water.

We now present a more general description of time-driven sensor networks. In time-driven networks, sensor nodes collect and report data from the physical environment periodically. The period between two consecutive data packets from a particular sensor node is referred to as the “data sampling frequency” or the “data reporting frequency”. Each sensor node typically sets an internal timer to manage the periodic data sampling and reporting. The data reporting frequency may be either static or dynamic. Applications that set a static data reporting frequency do so prior to network deployment to provide the user with data at fixed length intervals. In contrast, applications with a dynamic reporting frequency enable nodes to change their data reporting frequency during network deployment.

The periodicity of information flow in time-driven sensor networks provides for more predictable long-term behavior of the network. Designers can exploit the periodicity of information flow to optimize network performance. For example, an algorithm that is aware of the data reporting frequency can schedule transmissions of non-interfering nodes in such a way that maximizes spatial reuse and bandwidth utilization, while minimizing energy consumption (see Ch. 9). Time-driven sensor networks represent a simple model in which the nodes mostly report data and perform minimum data processing. Most of the processing takes place once the data reaches the data sink, which typically includes a larger computer with more resources. This class of sensor network applications requires minimal processing and memory resources at individual nodes. However, some applications require smarter sensor nodes that are able to partially process sensed data and to make appropriate decisions autonomously. Such applications allow the nodes to respond to events in the environment or to user queries.

Event-driven Sensor Networks

Event-driven data dissemination is a reactive model that closely fits the requirements of many sensor network applications. Nodes in event-driven sensor networks do not send their data periodically. Instead, nodes can adjust their data reporting behavior based on certain network events. Events are defined by setting thresholds for the sensed values. Once a sampled value exceeds a particular threshold, the node determines that the associated network event has occurred and it adapts its behavior accordingly.

We illustrate the operation of an event-driven sensor network through an example. Consider a network of sensors that is deployed for monitoring seismic activity along a fault line, as an early warning system for earthquakes. Earthquakes may occur several years apart, so the network should provide long-term unattended operation. Time-driven monitoring of the fault line is not a suitable choice in this application. Setting a data reporting frequency that is too high would deliver high data granularity, but it would also waste valuable energy resources for frequent transmissions, causing the nodes to deplete their batteries quickly. A small data reporting frequency would solve the network longevity problem, but the data granularity may be too low for reporting relevant seismic events. Instead, the nodes should only report worthwhile data, which they can determine based on thresholds for sensed physical data. For example, if a node detects that vibration has exceeded the threshold value, it determines that a seismic event is taking place and it increases its data reporting frequency for the duration of the event. Other nodes in the vicinity of the seismic activity follow the same behavior, while nodes that are not aware of the seismic event do not change their behavior.

As the above example shows, nodes in event-driven networks can assume different roles during deployment. The application must provide specifications of possible roles of nodes, and the network must support and adapt dynamic node roles. In particular, role-aware network mechanisms can use the cross-layer role information to optimize their configuration.

In many event-driven sensor networks, such as networks to detect forest fires or intrusions, data must arrive as quickly as possible to the user, so low latency is key for event-driven sensing applications. The event-driven model is a suitable reactive adaptation of the time-driven model, in which events in the deployment area drive data reporting. However, the event-driven model does not fit applications that require user control over network behavior.

Demand-driven Sensor Networks

While events in the operating environment drive data reporting in event-driven sensor networks, demand-driven sensor networks enable network entities, such as the end-user or software components within the network, to query the nodes for sensor data. Queries may be either name-based or attribute-based. Name-based queries specify the address or name of a particular set of sensor nodes from which data is required. Instead of targeting explicit sensor groups by name or address, attribute-based queries specify the attributes of the node targeted by the query.

We now present a brief example of a demand-driven network. Consider a network of wireless sensor nodes for traffic monitoring. Sensors are deployed below the tarmac on the highway to report traffic speeds and to infer traffic congestion. A user could issue the following query: What is the traffic on Interstate 405 at the Atlantic Avenue exit? The network propagates the query towards the sensors at Atlantic Avenue, which sample their sensors and report the data back to the user. This is an example of a name-based query. Alternatively, the user could submit the following attribute-based query: What are the points of congestion on Interstate 405 between Atlantic Avenue and Supelvida Boulevard? Only the nodes that detect slow traffic speeds in the area between Atlantic and Supelvida report their data in response to this query.

The data dissemination models for sensor networks are not mutually exclusive and they can be combined to provide a richer set of data dissemination options. For example, combining the time-driven and event-driven models can allow all nodes to report data infrequently, and nodes that detect an interesting event can report data with a higher frequency.

6.2.2 Application Performance Metrics

A single entity generally deploys a sensor network to monitor a physical space. Network users are concerned with the high level application performance of the network. Common requirements for sensor network applications include reliability, system lifetime, data freshness, and data resolution. The remainder of this section explores the application layer performance metrics for sensor networks in relation to the application classes in Section 6.2.1.

Reliability

Many sensor network applications require reliable data delivery. For example, a sensor network that monitors toxic leaks in an industrial facility must ensure without any doubt that any leak event is reported. From an application's perspective, reliable data delivery guarantees that transmitted data is received at its intended destination. Providing reliability to sensor network applications requires additional control messaging, such as handshakes and acknowledgements, at lower layers in the communications stack.

Network Lifetime

The deployment area of sensor networks may not be readily accessible. Military sensor networks may be deployed in hostile territory to monitor troop movements. Environmental sensor networks might monitor physically harsh environments, such as rainforests, volcanoes or deep ocean waters. Networks that monitor the structural integrity of bridges may embed the sensors within the concrete, making it very difficult to reach the sensors.

Because of the inaccessibility of deployment areas of many sensor networks, it is desirable to prolong the network lifetime to avoid replacing or recharging the sensor nodes frequently. Maximizing the network lifetime often requires energy efficient mechanisms at several lower layers across the network stack, such as support for adaptive duty cycles, power-aware routing, and adaptive transmission powers.

Delay

Delay is an important metric for some sensor network applications, such as real-time monitoring sensor networks, emergency response networks, target tracking networks, or industrial automation networks. Delay in sensor networks is typically intermittent and highly variable due to sensitivity of internode communications to environmental changes.

There are two delay components from a sensor network application's perspective: data freshness and response time. Data freshness indicates how recent the reported data is. The more recent data typically reflects the current state of the deployment area more accurately. The response time indicates the network application's capability to respond to environmental events or user queries within a given interval of time. For instance, a military application might set tight constraints on query response time to ensure that friendly troops can react quickly enough to enemy movements.

Data Resolution and Detection Probability

The detection probability refers to the probability that the sensor network can detect the occurrence of a relevant event within the deployment area. The data resolution refers to both the spatial and temporal granularity of the sensed data, which depends on the number of sensors that can monitor a particular point in the deployment area and on the data reporting frequency respectively.

Both the detection probability and the data resolution depend on the method and manner of sensor node deployment. The density of sensor nodes plays a major role for event detection. Naturally, sensor density involves a tradeoff with network cost and scalability. The network topology should be dense enough to ensure a sufficient data resolution for any relevant event within the deployment area. On the other hand, network designers must limit sensor density to minimize the scale and the monetary cost of the network.

The sensor deployment strategy also affects detection probability and data resolution. Sensor nodes may be regularly spaced within the deployment area to provide uniform data resolution. Alternatively, the network user may place more sensor nodes in a certain region of particular interest within the deployment area to provide higher data resolution in that region. A related issue is the accessibility of the deployment area. If the network user can deterministically deploy the nodes in the deployment area, then any deployment strategy will do. However, the deployment area may not be readily accessible. For example, a military application may require sensor deployment in enemy territory, in which case the sensors are dropped over enemy territory from a plane. Obviously, such a deployment strategy does not allow deterministic sensor placement. Instead, the sensors can be dropped at roughly equal distances in order to ensure with some probability that the network can detect relevant events.

Node mobility and node failures are two factors that contribute to dynamic network topology, affecting detection probability and data resolution. Reasons for node failures include unreliable sensor node hardware and the limited node battery resources. Thus, it is highly likely that some sensor nodes will fail during a long-term deployment. Sensor density and placement strategies should consider the expected node failure rate prior to deployment to ensure both acceptable data resolution and detection probability. Node mobility also affects the detection probability in a sensor network. The movement of sensor nodes may cause an initially acceptable network topology to provide an unsatisfactory level of detection probability and data resolution. In mobile sensor networks, the application may have to specify acceptable ranges for detection probability and data resolution. The acceptable range would depend on the degree of mobility and the maximum speed of the nodes.

Part II

Cross-Layer Approaches

7

Cross-Layer Design

This part of the book focuses on cross-layer design for ad hoc and sensor networks. In this chapter, we build up the case for cross-layer approaches through an examination of the unique requirements of ad hoc and sensor networks, and we outline guidelines for cross-layer models for these networks. The next two chapters focus on existing cross-layer approaches. Chapter 8 examines proposed cross-layer architectures for ad hoc and sensor networks, while Chapter 9 focuses on applied cross-layer approaches.

This chapter is structured as follows. Section 7.1 provides a comprehensive definition of cross-layer design approaches. Section 7.2 presents existing work on cross-layer approaches for traditional networks. Section 7.3 motivates cross-layer design for ad hoc and sensor networks. Finally, Section 7.4 presents the cross-layer design guidelines for ad hoc and sensor networks.

7.1 Cross-Layer Design: A Definition

Layered communication approaches typically separate communication tasks into several layers, with a clear definition of the functionality of each layer. In a layered communication stack, interaction among layers occurs through well-defined standardized interfaces that connect only the neighboring layers in the stack. Figure 7.1 depicts the traditional layer interaction through standardized interfaces in a strictly layered communication stack.

In contrast, cross-layer approaches attempt to exploit a richer interaction among communication layers to achieve performance gains. Srivastava and Motani [139] have recently proposed the following definition for cross-layer design: “Protocol design by the violation of a reference layered communication architecture is cross-layer design with respect to the particular layered architecture.” The basic premise of this definition is that cross-layer protocol design by definition violates the strictly layered interfaces of a reference layered communication architecture. However, Srivastava and Motani’s definition does not encompass all of the existing cross-layer approaches. For example,

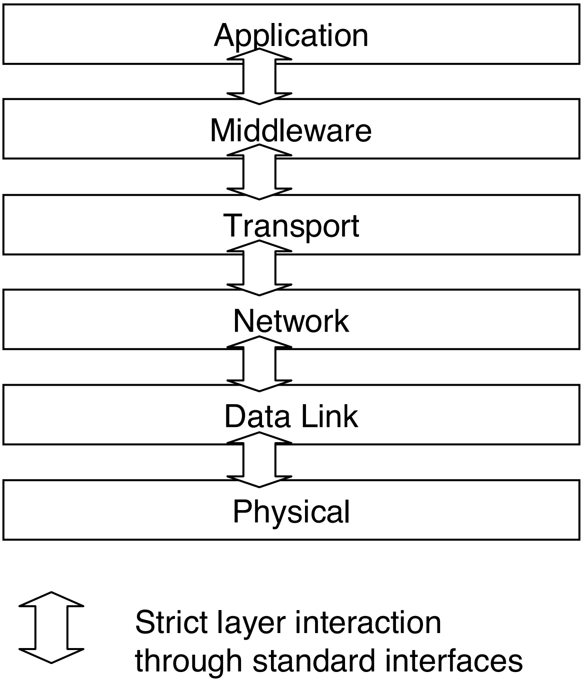


Fig. 7.1. A strictly layered communication stack

Chapter 8 examines proposed cross-layer architectures for ad hoc networks that maintain all the interfaces in a layered architecture intact while providing richer interaction possibilities among the layers. Another limitation of Srivastava and Motani's definition is that it considers only protocol design, whereas Chapter 9 and Chapter 8 reveal that algorithms and architectures can also adopt cross-layer design.

As a more comprehensive definition of cross-layer design, we propose a modified definition of cross-layer design approaches.

Definition of Cross-Layer Design: Cross-layer design with respect to a reference layered architecture is the design of algorithms, protocols, or architectures that exploit or provide a set of interlayer interactions that is a superset of the standard interfaces provided by the reference layered architecture.

The interlayer interactions fall into one of two broad categories:

1. **Information sharing**: Adjacent or non-adjacent communication layers can share information through new interfaces that are either unidirectional or bidirectional. Alternatively, a cross-layer architecture may support comprehensive state variables that are accessible to all communication layers [100,147,149].

2. **Design Coupling**: Design of new protocols or algorithms may try to exploit the features or avoid the weaknesses of existing mechanisms [159, 172]. More extreme cross layer approaches can partially or completely integrate the functionality of adjacent layers [160].

With this formal definition of cross-layer design, we proceed to follow the historical development of cross-layer design approaches in the next section.

7.2 Cross-Layer Design for Traditional Networks

Cross-layer design has its roots in infrastructure wireless networks. Early cross-layer models for wired networks have shown the benefits of cross-layer protocol interactions [136,137]. One of the first cross-layer proposal addresses TCP's failure in distinguishing network congestion from communication errors, which leads to invocation of the congestion control mechanism for both cases, as Section 5.1.1 discusses. This drawback of TCP is more pronounced for wireless networks, even infrastructure wireless networks, where the wireless channel exhibits both short-term and long-term variations [141]. To remedy this problem, future versions of TCP are expected to support an Explicit Congestion Notification [142] (ECN) bit. Routers can set the ECN bit to indicate that packets have experienced congestion. When the destination receives packets with the ECN bit set to one, the destination notifies the sender of the congestion so that the sender can reduce its transmission rate. Because the network layer at an intermediate node sets the ECN bit and transport layer at the destination reads it, the ECN bit represents a cross-layer interaction.

Another example is the enhancement of the Mobile IP hand-off with link layer information. Hand-off in Mobile IP networks occurs on detection of network changes at the network layer. Because network layer detection may be too slow, Sanmateu et al. [138] propose the use of signal strength information from the physical layer on the active links to reduce the hand-off latency.

The majority of cross-layer optimizations for traditional networks adopt an information sharing approach. Information sharing approaches involve minor modifications to existing models, and they can yield significant improvements in performance. Existing literature [139,143] on cross-layer design for infrastructure wireless networks has reached a consensus on the benefits of rich information sharing and interaction among layers. While design coupling may provide for deeper optimizations [139], it also moves further away from the layering models. As a result, design coupling trades off the performance gains for almost all of the interoperability and modularity benefits of layering. Design coupling has so far proved useful for optimizations that address particular scenarios [160] or specific technologies [172].

In this ongoing tension between the generality benefits of layered designs and the efficiency gains of cross-layer designs, a middle ground approach that seems to balance the tradeoffs emerges. The middle ground approach essentially adopts information sharing in an attempt to preserve the current layered models while enhancing them with richer interactions among layers to optimize performance. By preserving layering, the models keep the interoperability benefits. Support for richer interactions enables a closer coordination between the communication layers. In Chapter 8, we examine several such architectures for both ad hoc and sensor networks that adopt this middle ground approach. Before discussing these proposed cross-layer models for ad hoc and sensor networks, the next section motivates cross-layer design for these networks.

7.3 Why Cross-Layer Design for Ad Hoc and Sensor Networks?

This section investigates the factors that have led researchers to consider cross-layer design for ad hoc and sensor networks. We begin this section with an analogy between government interactions and data communication as a simple introductory motivation for cross-layer design. Next, we formally present the challenges that have motivated cross-layer design for ad hoc and sensor networks.

7.3.1 An Analogy

In this example, we draw an analogy between the interaction within a government structure and the interaction within a network. Consider first how highly centralized governments, such as the government of France, operate. Most, if not all, of the major executive decisions are taken by the central government, which has a well-defined hierarchy of employees. The bureaucracy specifies well-defined channels through which government employees at each level in the hierarchy interact with lower-level employees, higher-level employees, and possibly with citizens. For example, a mid-level manager in the ministry of health can give directives and receive reports from employees under him that directly interact with the citizens, and he can create reports for and receive directives from the higher-level manager. Within this structure, the higher level government officials and the citizens have no direct interaction. The employee hierarchy in this example resembles layered architectures that define standard interfaces between neighboring layers of the communication stack and that disallow communication between non-neighboring layers. For instance, in traditional layered models, the network and transport layers have no direct interaction with the physical layer.

Now consider the other extreme of government represented by the direct democracy in a federated government. Take Switzerland, for example, which has a federated government of three cantons that applies the concept of direct democracy to hold elected officials immediately accountable for all their actions. One factor that has strengthened direct democracy in Switzerland is the federated government system, which allocates far more power to the regional canton governments than the federal government. Shifting the government administration activities to the local governments enables a richer interaction between the people and the government officials, as the citizens can keep track of and interact meaningfully with elected officials in smaller districts. In fact, the richer interaction between the constituents and the government becomes crucial to the government efficiency with the increased distribution of governance activities. The smaller tax revenues of local administrations highlight the need for making efficient use of these resources, as any waste of tax revenue will be directly felt by the citizens. To complement the highly federated government model, Switzerland has the most prominent example of direct democracy in the world. In the past 120 years, the system has enabled Swiss citizens to vote on 240 referendums on key issues [140]. The direct democracy system also enables citizens to frequently contact and select elected officials, and to hold them accountable for all their actions while in office.

In our analogy, the local governments in the Swiss system represent nodes in ad hoc networks, and the citizens and government officials at different levels represent the protocol layers. Two concepts from the federated direct democracy system are relevant for ad hoc and sensor networks: distributed decisions and richer interactions. The distributed decisions that take place in the federated government model resemble the distributed network management decisions by individual nodes in ad hoc and sensor networks. Just like in the Swiss system, the efficient performance of the node may benefit, and in some cases may even require, richer interactions among the communication layers of the node in the network. Cross-layer design for ad hoc and sensor networks aims to provide richer interlayer interaction within a node to improve the node's performance.

7.3.2 Motivating Factors

Having motivated cross-layer design through the government analogy, we now formally present the factors that motivate cross-layer design for ad hoc and sensor networks. We will first examine the motivating factors for general ad hoc networks, followed by factors that are specific to sensor networks.

The motivating factors for cross-layer design for ad hoc networks include:

1. **Cross-Layer Aspects**: Nodes in ad hoc networks must manage several performance aspects, such as power management, system management, security, and discovery that cut across traditional layers. For instance, both medium access and routing decisions have significant impact on power consumption, and the joint consideration of both can yield more efficient power consumption. The strict separation of layers and standardized interlayer interfaces in traditional approaches do not enable sufficient interaction among layers to make joint decisions to optimize these cross-layer

aspects. This has led to the proposal of new interaction models to support cross-layering, ranging from a more relaxed information flow and sharing between layers to full-fledged merging of layer functionalities.

2. **Distributed State**: Unlike traditional infrastructure models where base stations have a global view of network state, the network state in ad hoc networks is generally distributed across the nodes. Each node forms its own local view of state, representing a partial view of the overall network state. In most cases, it is not feasible to collect network state at any one node, which prevents the use of any centralized optimization algorithms. As such, each node can run distributed algorithms locally using its partial view of network state. Distributed algorithms can exploit a cross-layer design to enable each node to perform fine-grained optimizations locally whenever it detects changes in state.
3. **Mobility**: Mobility introduces an additional challenge for ad hoc network design. Imagine if all the Internet routers were mobile. Routing protocols would have to cope with this mobility by constantly adapting routing state to the changing router positions. Now consider mobility in the context of ad hoc networks, where no node has global view of network state. Mobility management poses an additional challenge to the battery-powered nodes in ad hoc networks, which have to adapt their behavior to the changing node locations. Mobility causes changes for the physical layer (e.g. interference levels), the data link layer (e.g. link schedules), the routing layer (e.g. new neighbors), and the transport layer (e.g. connection timeouts). As such, a cross-layer design enhances a node's capability to manage its resources in mobile environments.
4. **Wireless Link Properties** Wireless links are more susceptible than wired links to interference variations and channel errors. One classic example is the TCP congestion control problem over wireless links in which TCP misinterprets a packet loss due to channel error as a sign of congestion. Wireless links are also more vulnerable to security attacks because of easy access to the wireless channel. Providing higher layers with awareness of the wireless link status enables nodes to adapt their configuration better to physical layer properties. For example, a routing protocol that detects a drop in quality of a particular wireless link can create a new route to divert traffic to another wireless link.
5. **New Communication Modalities**: Ad hoc network design can exploit the broadcast nature of the channel to enhance performance. For example, nodes can snoop on neighboring transmissions in order to evaluate the quality of links with neighbors. Antenna arrays also enable the reception of multiple packets simultaneously on the wireless channel. Data packets corresponding to several connections could arrive simultaneously at a node. The close coordination of the routing, data link, and physical layer can ensure the timely forwarding of data for all the connections. The broadcast nature of the wireless channel also provides fertile ground for supporting multicast interactions.

6. **Inherent Layer Dependencies** Several interlayer dependencies motivate cross-layer design for ad hoc and sensor networks. The data link and routing layers in ad hoc networks exhibit both variable and algorithmic interaction [144], suggesting the need for design coupling of these layers. The data link layer is also closely related with the physical layer. If provided with current channel conditions, the data link layer can adapt error control mechanisms in a dynamic manner, thereby improving throughput [145].
7. **Resource-Constrained Nodes** The form factor of mobile nodes for ad hoc networks keeps decreasing, imposing the use of smaller batteries for these nodes. The relatively slow improvement in battery technology implies that the nodes must interact in an ultra-efficient manner to maximize the lifetime of the battery. Cross-layer design approaches can expose power related variables at several layers, enabling nodes to efficiently utilize their energy resources.

The above factors apply to both general ad hoc networks and sensor networks. The following additional factors characterize cross-layer design for sensor networks:

1. **Global Performance Guarantees**: Nodes in sensor networks typically collaborate to achieve a common network goal. As such, sensor networks must support the enforcement of performance guarantees on a network-wide scale. The absence of a central controller that has a global view of network states requires that the distributed algorithms running at the nodes adhere to the global performance policies. Therein lies the main challenge in the design of communication protocols for sensor networks. Global performance policies can only be enforced through local decisions made at individual nodes in a greedy and distributed fashion. Efficient network operation requires that the local algorithms that optimize global performance adapt to the local state at each node. The local algorithms should strike a balance in adapting to the dynamic local node states, while adhering to the global optimization policy.
2. **Application-Specific Policies**: The semantic definition of what constitutes “good performance” varies greatly between different sensor network applications. For example, a sensor network that monitors pollution levels in a watershed [14] prioritizes network longevity to avoid frequent retrieval and recharging of node batteries. In contrast, a sensor network for toxic leak monitoring in a factory emphasizes reliability and delay to guarantee that any leak event is reported promptly. The wide range of possible sensor network applications and the application-specific performance requirements of each application suggest the need for a general model that is tunable to each application. The general model should ensure that the relative impact of quality of service metrics, which inherently cut across layers, can be tuned to suit particular applications.

3. **Platform-Specific Properties**: Because they interface the physical world with the digital domain, sensor network operation inherently depends on the hardware platform that performs the sensing and communication. While the specifications of each sensor network platform vary widely, the platforms must provide higher layers and applications with sufficient control over sensing and communication behavior. Successful design strategies should hide the differences of various hardware platforms from higher layers (for instance through hardware abstraction layers) in order to enable efficient mapping of application performance goals to the physical layer parameters. Cross-layer models provide a richer interaction set for mapping the abstract application performance requirements to the physical platform-specific properties [149].
4. **Unattended Operation**: Most sensor network applications require unattended operation, so the network nodes must independently determine how to react to mobility, loss of connectivity, or energy depletion. Cross-layer approaches provide nodes with unprecedented visibility about their current resources and processes, which contributes favorably to node autonomy and self-configuration.

This section has identified the factors that motivate cross-layer design for ad hoc and sensor networks. The next section builds on these factors to develop general guideline for cross-layer design in ad hoc and sensor networks.

7.3.3 Design Challenges

Cross-layer design also involves risks that, some researchers would argue, may outweigh its benefits. Kawadia and Kumar [146] emphasize the ongoing tension between performance and architecture: while cross-layer design offers short-term performance benefits for a particular system over traditional architectures, it also limits the modularity and interoperability offered by architectures. They identify the following challenges for cross-layer design:

1. **Unstructured Code**: The implementation of several cross-layer design optimizations within a system may lead to spaghetti-like code that is unstructured and thus difficult to maintain. The unstructured code may stifle innovation as it makes it harder to modify or upgrade existing systems. It also raises questions on the proliferation and longevity of the system. Finally, the unstructured code could eventually lead to an increase in per-unit cost. All of these factors can be regarded as *long term* performance metrics that may be adversely affected by cross-layer design.
2. **Multiple Interactions**: Cross-layer design opens the floodgate of information flow across layers, raising concerns on multiple, sometimes subtle, interactions among existing layers. For example, a cross-layer approach may determine that current conditions necessitate configuration A at the routing layer and configuration B at the MAC layer, although configurations A and B may have unknown interactions that lead to degraded

network behavior. A major challenge of cross-layer design is the clear identification and exploration of the possible dependencies and interactions among the system processes at different layers.

3. **Short-term Benefits**: According to Kawadia and Kumar, cross-layer design offers performance benefits for a particular system, yielding short-term gains. In contrast, an architecture offers a model for sustained innovation in a system, so it offers long-term gains. Their contention is that the long-term gain of architecture overweighs the short-term gain of cross-layer optimizations. However, many ad hoc and sensor network applications are quite specific in nature, so the short-term performance gains of cross-layer design may be far more important for the network user to make efficient use of scarce node resources.
4. **Holistic Perspective**: Because cross-layer optimization involves dependencies among other system processes and other systems, cross-layer designers must consider the impact of their design with a holistic view that includes the long-term development and innovation considerations.

The above challenges suggest that the benefits and opportunities for improving performance through cross-layer design should be embraced cautiously through preservation of a form of architectural framework to provide modularity, define interactions, and drive innovation. The next section attempts to outline the guidelines for cross-layer design that can balance these design tradeoffs between architecture and performance.

7.4 Cross-Layer Design Guidelines

The discussion in the previous sections yields design guidelines for ad hoc and sensor network cross-layer design. In this section, we examine the guidelines for efficient and deployable cross-layer models for ad hoc and sensor networks.

7.4.1 Compatibility

The quick development of the Internet has shown the importance of an architecture in ensuring interoperability and compatibility among users on a global scale. Although development of new protocols has skyrocketed recently, ad hoc and sensor networks need a reference architecture [147, 149] within which protocols developed by one research group can interoperate with other development efforts. In addition to providing compatibility, a successful reference architecture must enable flexible cross-layer interaction to support fine-grained optimizations.

Backward compatibility with IP-based networks is another consideration for designers of cross-layer approaches for ad hoc and sensor networks. The pervasiveness of Internet-enabled nodes provides a fertile gateway through which ad hoc and sensor network nodes can communicate globally. Many ad hoc networks, such as networks of laptops and PDAs, can provide each node an individual IP address for directly interfacing with other Internet nodes. In sensor networks, nodes may rely on a gateway node, usually the data sink, for an interface with the Internet.

7.4.2 Richer Interactions

Cross-layer design must address two types of interactions: interlayer interactions; and internode interactions. The interlayer interactions deal with the allowable information exchange between communication layers within each node. Cross-layer approaches must specify the implementation details of the interlayer interactions. For example, providing a comprehensive set of network status variables uses a shared database model that is accessed by all layers [147]. Another possible implementation is creating new abstractions that depart from a layered approach [139].

Another class of interaction is the internode interaction. The sharing of information across layers enables each node to form a more comprehensive representation of its own state. In addition to knowledge of local state, most optimization goals can benefit from information about the state of other nodes in the network. In many cases, the quality-of-service requirements target global network performance rather than per-node performance, suggesting the need for global network optimization. Global network optimization is particularly challenging for ad hoc and sensor networks because of the lack of a central controller that can collect dynamic node states.

The crosscutting constraints represented by the need for global quality-of-service and the availability of local state information at each node requires a policy that balances the two constraints. The internode collaboration strategy in ad hoc and sensor networks will be inherently greedy because of its dependence on partial state information. However, all greedy decisions must be aware of the global QoS requirements and must try to optimize local node configurations to achieve the global performance requirements.

7.4.3 Flexible and Tunable

Within the wide and diverse application space for ad hoc and sensor networks, there are many quality metrics of potential interest: power consumption; network throughput; network lifetime; delay; delay jitter; reliability; fairness; and hardware cost. Different applications may emphasize specific quality metrics. For example, an ad hoc network for video transmission among several home entertainment devices requires strict throughput, delay and delay jitter guarantees. At the other extreme, an ad hoc sensor network deployed for environmental monitoring of the ocean or an agricultural field requires the longest possible network lifetime and minimal power consumption. Performance targets may also change during network operation. For example, a sensor network for real-time monitoring of forest fires may be deployed for months before any significant fire event occurs. During that time, the network nodes can send infrequent updates to report their data in order to maximize their expected lifetime. Once a fire occurs, the nodes that detect the fire can start reporting data more frequently to guarantee sufficient data granularity for the user.

The above examples expose the need for design flexibility at distinct stages: initial tuning at development time according to application and platform properties and real-time adaptation during network operation. Cross-layer approaches should provide a general model that is sufficiently tunable to support the wide and diverse range of applications and platforms of ad hoc and sensor networks. In addition, cross-layer approaches must enable dynamic tuning of network configuration during network operation. Interlayer interaction between protocols must allow the nodes or the network operator to fine-tune the QoS policy according to dynamic network conditions.

8

Cross-Layer Architectures

This chapter presents proposed cross-layer architectures for ad hoc and sensor networks. Cross-layer architectures provide a holistic view of the cross-layer model, focusing on how to interface or integrate layers from across the communication stack. Recent cross-layer architectures have promoted information sharing across existing protocol layers while maintaining the separation of functionality between layers. The domination of information sharing over design coupling in cross-layer architectures is not a coincidence. By definition, an architecture creates clearly defines entities and outlines the interactions between these entities. Because design coupling combines the functionalities of different layers, it essentially creates new superlayer entities. The superlayer entities are treated as layers that must interact with the remaining entities, leading back to the architectural issues. Information sharing, on the other hand, fits well within a well-defined architecture, since it only requires the specification of new interfaces or the expansion of existing interfaces.

Within this context, Table 8.1 provides an overview of cross-layer architectures for ad hoc and sensor networks. All the architectures in Table 8.1 adopt an information sharing approach. Among the surveyed models, two architectures are designed for ad hoc networks and three are designed for sensor networks. Only one of the architectures target both ad hoc and sensor networks. All but one of the architectures are platform independent. Interestingly, most architectures that address internode interactions rely on partial state information, ranging from local state, neighborhood state, to global state.

We now examine the proposed cross-layer architectures more closely based on their target network types: ad hoc networks; sensor networks; and ad hoc and sensor networks.

8.1 Ad Hoc Networks

Cross-layer architectures for ad hoc networks consider the need for distributed algorithms to support QoS guarantees in highly dynamic environments. This

Model	Network		Target	State		
	ad hoc	sensor		local	neighbor	global
MobileMan	X		platform-independent	X		
CrossTalk	X		platform-independent	X	X	X
Sensor Protocol		X	platform-independent	X		
TinyCubus		X	TinyOS	X	X	
Lu		X	platform-independent	X		
Jurdak	X	X	platform-independent	X	X	

Table 8.1. Cross-Layer Architectures

section examines 2 cross-layer architectures for ad hoc networks: (1) MobileMan; and (2) CrossTalk. The common feature of the two architectures in this section is the presence of a common repository of state information from all layers in the communication stack. The common repository provides a unified and standard interface for sharing state information across layers. The remainder of this section describes the two architectures in more detail.

8.1.1 MobileMan

MobileMan [147] proposes a cross-layer architecture for ad hoc networks that tries to balance the interoperability benefits of a layered architecture and the efficiency benefits of cross-layering. Figure 8.1 shows the proposed architecture of MobileMan.

The MobileMan architecture preserves the original layered architecture at its essence. The main contribution of MobileMan is the introduction of a Network Status component that functions as a repository for information collected by network protocols throughout the stack. The Network Status component stores all of the relevant variables, relating to energy management, security, and cooperation, to be accessed by protocols at different layers. MobileMan specifies a set of interfaces for cross-layer interaction that enables each layer in the layered architecture to access information in the Network Status component.

One attractive feature of MobileMan is that it separates the interaction of communication layers with the Network Status component from the traditional layer-to-layer interactions. Through this separation, network designers can create a new protocol at a particular layer and plug it into the MobileMan architecture. The only requirements of creating new protocols or redesigning existing protocols are providing the standard interfaces to neighboring communication layers and to the Network Status component.

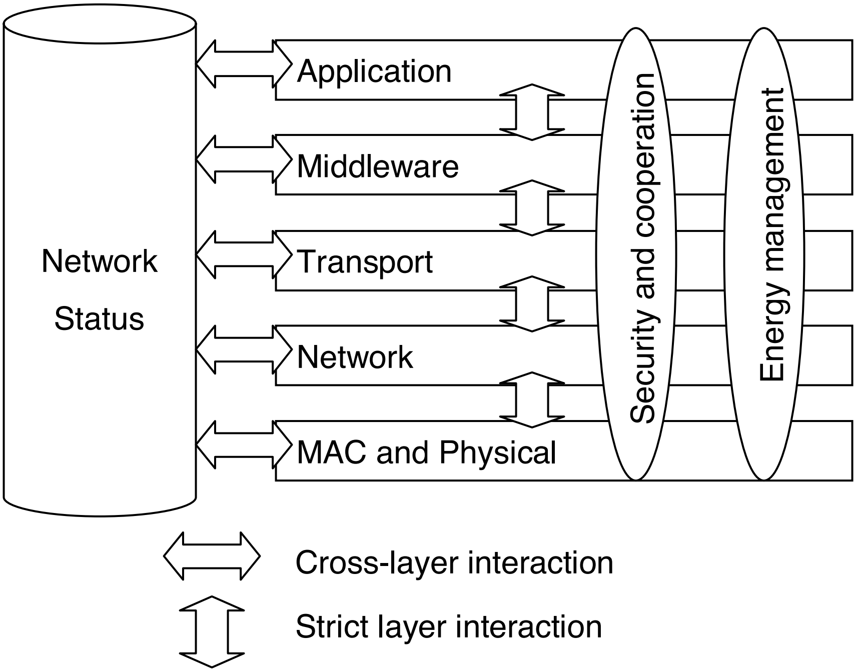


Fig. 8.1. Mobileman architecture [147] (©2004 IEEE)

8.1.2 CrossTalk

The CrossTalk [148] architecture for ad hoc networks also defines cross layering as the enhancement of traditional layered architectures through sharing of information between layers “which can be used as input for algorithms, for decision processes, for computations, and adaptations”. Figure 8.2 shows the proposed CrossTalk architecture. Note the resemblance of the cross-layer entity in CrossTalk to the Network Status component in MobileMan.

The CrossTalk architecture incorporates two views of network state: (1) a local view; and (2) a global view. The local view includes node-specific information compiled from the layers of the communication stack into the cross-layer entity. The global view is constructed from information gathered by CrossTalk’s data dissemination process, which operates as follows. A source node piggybacks its local information into its data packets, adding a small communication overhead. Note that only the source node includes its local information in data packets, and nodes that forward the packet do not add their local information. Through this data dissemination process, each node overhears local state updates from several nodes, enabling it to form its own partial view of global network state. Each arriving packet at a node N is tagged with a timestamp value and a distance value indicating the distance from the source node to node N. The main idea is to enable nodes to have a

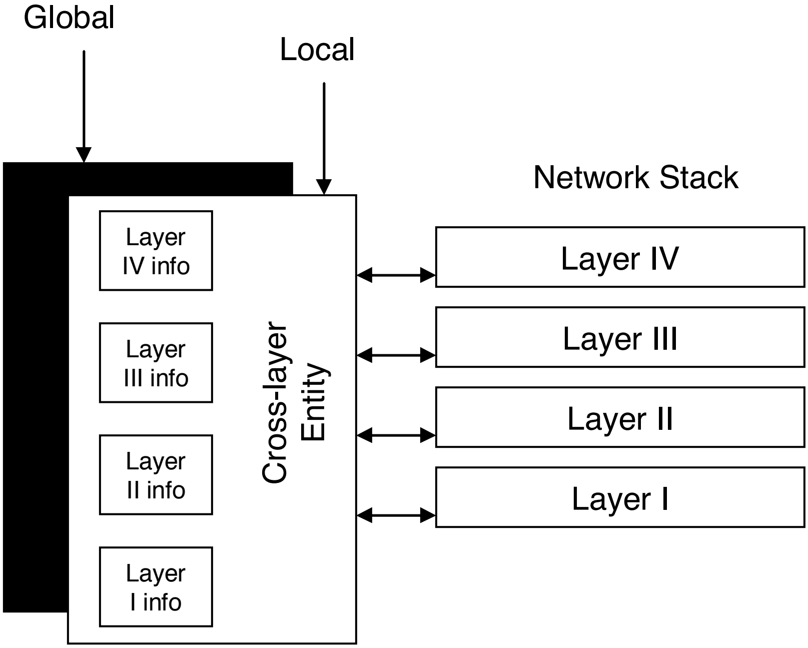


Fig. 8.2. CrossTalk architecture [148] (©2005 IEEE)

reasonably up-to-date and correct view of the network state. Nodes can then compare the global view with their local view to determine their behavior. For example, if a node's duty cycle is much higher than the network average, the node deems itself overloaded and makes the necessary adjustments.

CrossTalk weighs samples of state information according to time and distance. In particular, more recent samples are given higher weight since they better reflect current network state. Regarding distance, nodes give the highest weight to samples that arrive from intermediate distances. The rationale is that direct neighbors of a node may have similar local views to the node's own local view because of proximity. Nodes that are too far away may be at the edge of the network, so their state may not accurately reflect network state.

The main theme of CrossTalk, cross-layer optimization through a balance between local and global views, is also the main theme of Jurdak's cross-layer optimization framework, which is discussed in Section 8.3.1.

8.2 Sensor Networks

Sensor networks are highly application-specific, especially in terms of QoS requirements. There is also a wide range of sensor node platforms, ranging from tiny mote-class modules to more powerful microservers. As a result, a common aspect driving sensor network cross-layer architectures is the need to efficiently map application performance requirements with specific physical layer entities and with the rest of the communication protocol stack.

8.2.1 Sensor Protocol

The Sensor Protocol (SP) architecture [149] shown in Figure 8.3 independently proposes a sensor network cross-layer architecture that is similar to MobileMan and CrossTalk in its cross-layer aspect visibility. The SP architecture outlines the guidelines for establishing a sensor network architecture that enables interoperability among different components. Their recommendation is to have a Sensor Protocol (SP) abstraction layer, similar to the role of IP in the Internet, over which all new sensor network protocols and services could reside. Figure 8.3 shows the SP architecture.

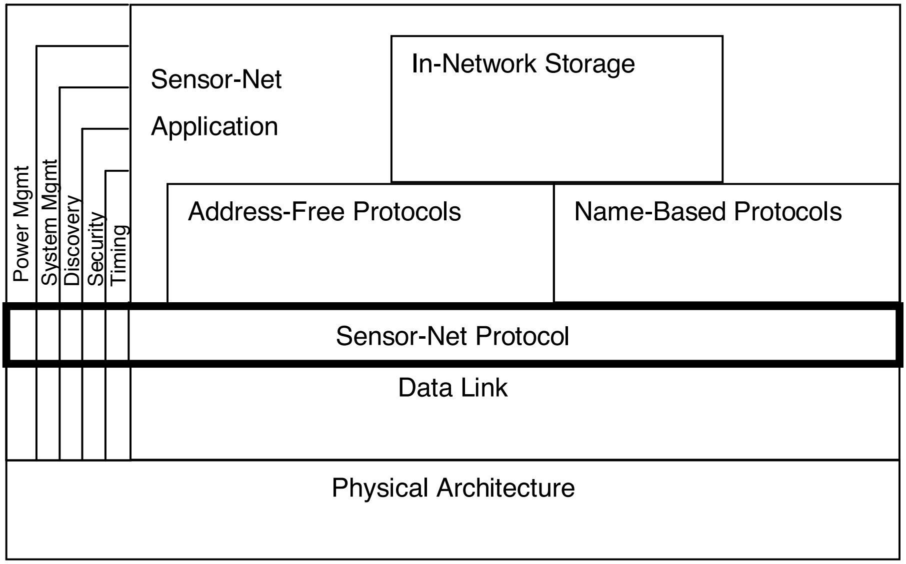


Fig. 8.3. SP architecture

The sensor protocol serves as the “narrow waist” of the architecture. All higher and lower layer protocols and services need only interface with the SP protocol. The architecture also proposes cross-layer visibility and management of several aspects, such as power management, system management, security, discovery and timing. In particular, the SP architecture specifies that all layers in the system should have access to these services. Culler et al. identify two crosscutting challenges for the design of SP: to provide an interface rich enough for application/SP collaboration, and to keep that interface platform independent. The former challenge stems from the need for customized performance for each application, while the latter challenge stems from the wide range of sensor platforms available.

To address these two challenges, the SP architecture proposes having platform independent power management interfaces that are customizable to the details and tradeoffs of specific platforms. The SP architecture also provides a rich two-way interface between SP and the other protocols. These interfaces enable the SP layer to share its information with other layers, and it also enables other protocols to share information with SP so that it can make system-wide resource management decisions.

8.2.2 TinyCubus

The TinyCubus [150] is a cross-layer framework for sensor networks based on TinyOS that aims at providing a flexible and adaptive infrastructure for optimizing sensor network applications with diverse performance goals. Figure 8.4 shows the main components of the TinyCubus architecture, which approaches cross-layer optimization with a programmatic and implementation perspective.

The TinyCubus architecture defines the Tiny cross-layer framework that provides interfaces to support information sharing among layers and to enable lower layers to invoke application-specific code. To support cross-layer information sharing, the framework employs a state repository that stores all relevant parameters for cross-layer access. To enable components to access the state repository, the model defines a specification language that is responsible for generating interfaces to connect components with the shared data. The invocation of application-specific code by lower layers (also referred to as callback) is achieved through an extension of existing TinyOS component wiring mechanisms.

Another component of the TinyCubus architecture is the Tiny Configuration Engine. The Configuration Engine includes a topology manager responsible for network self-organization and for assigning a role to each node, such as SOURCE, AGGREGATOR, SINK or CLUSTER HEAD. The Configuration Engine also enables code distribution through the network for reprogramming the nodes in-situ.

The final component of the TinyCubus architecture is the Data Management Framework, which provides a standard set of data management and system components and chooses the best component set based on three dimensions: system parameters, application requirements, and optimization parameters. The system parameters refer to factors such as sensor density, node resources, or mobility. The application requirements dimension refers to quality of service metrics such as reliability or delay. Finally, the optimization parameters determine the selected algorithm for the given constraints at the three dimensions.

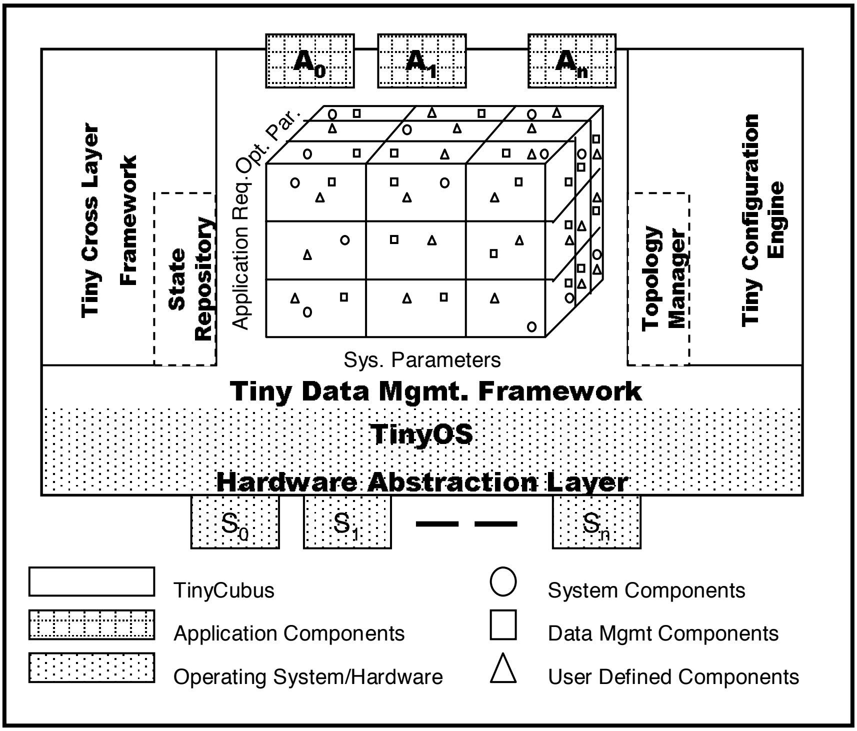


Fig. 8.4. The TinyCubus architecture [150]
(©2005 Oldenbourg Verlag)

8.2.3 Lu

The cross-layer framework of Lu et al. [162] attempts to formalize cross-layer dependencies and tradeoffs so that network designers can address these tradeoffs in a principled way. Lu's framework considers a reference architecture similar to both SP and TinyCubus, shown in Fig 8.5.

The reference architecture considers horizontal and vertical layers. Horizontal layers encompass traditional layer functionalities, while vertical layers represent cross-layer interactions. In addition to the traditional physical, MAC, network and transport layer, Lu's reference architecture introduces a horizontal connectivity maintenance layer between the network and MAC layers to maintain a connected network topology that withstands dynamic network states and intermittent connectivity. Lu's architecture also introduces a data management layer between the application and transport layers, which is responsible for data placement, data discovery, and in-network processing.

The two vertical layers in Lu's architecture provide sensor network specific services. The coverage maintenance layer guarantees a sufficient number of sensor nodes monitor a target area . The other vertical layer, called loca-

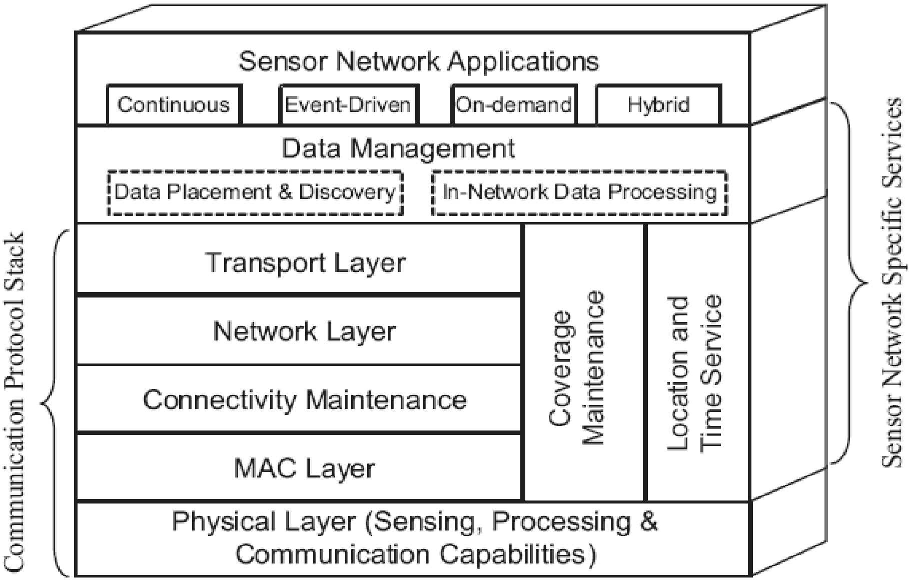


Fig. 8.5. Lu's reference architecture [162] (©2005 IEEE)

tion and time service layer, enables sensor nodes to determine their relative locations in the network and provides a means for synchronization among the nodes.

The model establishes dependencies between performance parameters at each communication layer. In particular, the main contribution of Lu's architecture is the mapping of application QoS quality requirements to parameters at all other layers in the architectures. The model adopts a top-down approach, as it starts with each application layer performance metric and progressively descends in the communication stack to identify the metric dependencies at each layer, reaching as far down as the physical layer.

8.3 Ad Hoc and Sensor Networks

The cross-layer architectures in this chapter have addressed ad hoc networks and sensor networks independently. This section presents a proposed framework for both ad hoc and sensor networks.

8.3.1 Jurdak

Jurdak [100] presents a cross-layer optimization framework for both ad hoc and sensor networks that also advocates full visibility of relevant state information among communication layers. The departure point of Jurdak's framework from most of the previous architectures is the specification of both interlayer and internode interactions with more detail.

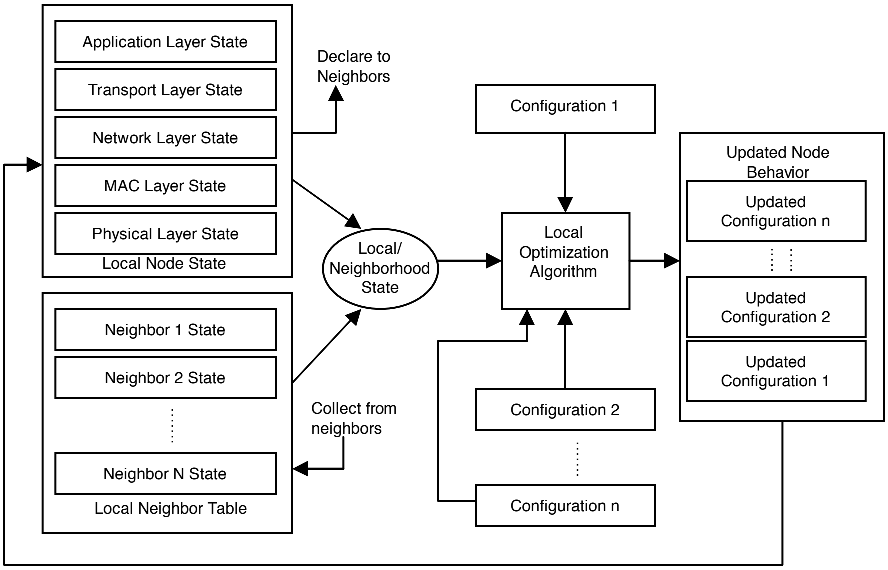


Fig. 8.6. Jurdak's general optimization framework

Figure 8.6 shows Jurdak's general framework. The framework's state definition provides the network designer with flexibility in specifying the relevant state variable, to ensure that the framework is tunable to the performance requirements of different applications. For example, an ad hoc network for video transfer requires throughput and delay guarantees. In contrast, long term monitoring sensor networks may require energy efficient behavior. These performance issues cut across layers, similar to the power management and system management aspects in the SP architecture, and they require internode collaboration.

In a similar approach to CrossTalk [148], the framework adopts a holistic and flexible definition of *local node state* which can consolidate state information from all layers of the communication stack, depending on the application. The flexible definition of node state in Jurdak's framework is similar in essence to the Network Status component in MobileMan, although the two implementations differ in form. Each node can declare its state information to its direct neighbors, and nodes maintain a state table of neighboring nodes, which is denoted as the *local neighbor state*. The combination of the local and neighbor state at each node comprises the overall node state. Each node uses its overall state as input to a locally resident optimization algorithm. The other inputs to the optimization algorithm include a set of local configurations at the node. The local optimization algorithm determines the resulting node behavior consisting of the updated configuration set. The updated configuration may result in changes to the local node state. The strength of Jurdak's approach is its validation for three different case studies of ad hoc and sensor networks, which we present in detail in Part III of the book.

9

Applied Cross-Layer Approaches

The previous two chapters have presented cross-layer design and architectural issues for ad hoc and sensor networks. This chapter focuses on the applied cross-layer approaches for ad hoc and sensor networks.

Applied cross-layer approaches for ad hoc and sensor networks fall into two broad categories: design coupling approaches and information sharing approaches. Design coupling approaches ignore layer boundaries to propose algorithms that optimize network performance metrics by integrating functionalities from different layers. Information sharing approaches attempt to share information across layers while maintaining architectural protocol boundaries.

Based on our survey of existing approaches, we extract 5 characterizing features of applied cross-layer approaches for ad hoc and sensor networks: (1) global performance goals; (2) target networks; (3) independent optimization variables; (4) dependent optimization variables; and (5) implementation design. The first feature, *global performance goals*, deals with the desired performance improvement of the cross-layer model on a global or network-wide scale. Next, the *target networks* feature specifies the class of networks for which a cross-layer model is customized. The third feature, *independent optimization variables*, includes one or more variables that serve as the input of a cross-layer model. Supporting the cross-layer nature of the models in this chapter, the independent variables may provide state information from several layers across the network stack. The model can then use the independent variables for determining how to optimize node configurations, which leads us to our fourth feature. A model's *dependent optimization variables* represent the output set of variables from across the communication stack. In other words, the model determines the dependent optimization variables based on the synthesis of the independent variables and the current configuration to determine optimal network behavior. Our final feature deals with a models' proposed implementation approach. Some models are algorithmic in nature, while others explicitly specify the internode collaboration strategy and the locale of required computation. Centralized models collect network state information at a central controller that runs comprehensive optimization algorithms, while distributed models spread the required computation for cross-layer optimization across several nodes.

Before classifying the cross-layer approaches according to the manner in which they address each feature in the subsequent sections, the next two sections provide an overview of the operation of each cross-layer approach. Section 9.1 discusses the optimization approaches that rely on design coupling. Section 9.2 presents cross-layer approaches that focus on information sharing within an architectural preserving framework. The remaining sections compare the applied approaches according to the 5 characterizing features. Section 9.3 surveys and compares the different performance goals of applied cross-layer approaches. Section 9.4 classifies the cross-layer approaches according to their target networks. Section 9.5 discusses the independent optimization variables considered in the existing approaches, and Section 9.6 reveals how each model uses the independent variables for determining optimal configurations. Section 9.7 focuses on the practical and architectural implementation issues of each model in distributed ad hoc and sensor networks.

9.1 Design Coupling Approaches

Cross-layer approaches that rely on design coupling of adjacent layers disregard layer boundaries and treat network mechanism as a single algorithmic block that optimizes certain performance metrics. The combinatorial aspect of these approaches typically yields a lack of specification of the functional or architectural issues, focusing mostly on algorithmic design.

Table 9.1 provides an overview of the features of existing cross-layer approaches for ad hoc and sensor networks that rely on design coupling. The columns in Table 9.1 correspond to the 5 features, while the rows correspond to existing approaches. Sections 9.3 through 9.7 will revisit the classification in Table 9.1 with more detail.

9.1.1 Girici and Ephremides

Girici and Ephremides [158] propose a cross-layer approach based on joint routing and link scheduling for ad hoc networks. Their approach aims at making routing and link activation decisions that are aware of three aspects: energy, delay, and network lifetime. To enable aspect-aware decisions, the nodes store the state of neighboring nodes, including residual battery energy, available number of transceivers, and transmission power requirements of each neighbor.

The purpose of this approach is to solve the following energy efficient routing problem: to enable a node to select the best next hop for each packet according to energy, delay, and network lifetime considerations. For this purpose, Girici and Ephremides's approach introduces a new link cost metric that considers three terms: transmission energy per packet (physical layer aspect),

Model	Goals			Target	Input Aspects			Optimizations				Design
	1	2	3	Network	Network	Data Link	Physical	Transport	Network	Data Link	Physical	
Girici [158]	network life	minimum delay	energy efficiency	TDMA ad hoc networks	queue length	link throughput	trans. power utilization		route cost	link schedule		unspecified
Cruz [159]	energy efficiency			wireless base stations		average link rate	maximum trans. power		route cost	link schedule	trans. power	distributed
ElBatt [160]	maximum throughput	energy efficiency		TDMA CDMA ad hoc networks			SNR interference			link schedule	trans. power	centralized
Kozat [161]	energy efficiency			TDMA ad hoc networks		link rates	BER SNR			link schedule	trans. power	centralized
Lu [162]	maximum throughput	minimum delay	energy efficiency	TDMA sensor networks			SNR			link schedule	trans. power	unspecified
Madan [163]	network life			TDMA sensor networks	route flow		SNR			link schedule	trans. rate power	unspecified
Wang [166]	maximum throughput	fair access		ad hoc networks		link rate		session rates		link attempt probability		distributed
Cui [164]	energy efficiency			TDMA sensor networks			processing/ trans. power		route cost	link schedule	MQAM trans.	unspecified
Merz [172]	maximum throughput			UWB ad hoc networks			TH-code			timers	code rate	distributed

Table 9.1. Applied Cross Layer Approaches based on Design Coupling

expected volume on a node's links throughout the network lifetime (data link layer aspect), and the queue length along the directed link (network layer aspect). The link metric definition provides a tunable coefficient for each term in order to favor any of the three terms.

This approach adopts a slotted time structure with a separate time-slotted control channel that enables nodes to reserve slots for data delivery. Each time slot is subdivided into a request and a confirm slot. In order to request a time slot, a node issues a REQUEST message, to which the receiver replies with a CONFIRM message. Upon receiving the CONFIRM message, the sender can proceed with data transmission in the reserved time slot.

Girici and Ephremides's approach attempts to schedule links by considering communication performance as a utility and assign each link a dynamically changing utility value, according to a predefined link activation utility metric. The ultimate goal is maximizing the sum of the utility values of the links in the resultant set of scheduled links.

The new link utility metric encourages the activation of congested links, and it discourages activation of links with high-energy destination nodes. Through this mechanism, the queue sizes of the highly congested links remain large and they are not preferred in routing, which prevents those nodes from early death. The new metric also encourages activation of links requiring low RF power, so that their queue sizes remain low and they are preferred for routing.

The performance evaluation of this approach through simulations to investigate the effect of each term the in routing metric confirms that the average energy per packet in the network is minimized when the routing metric only includes the transmission energy per link. Expanding the routing metric to include delay or lifetime metrics causes higher transmission energy per packet. Another intuitive result is that delay performance is best when queue sizes are included as part of the routing metric. Finally, including the node residual energies into the routing metric enables more load balancing in the network and it prolongs the time at which the first node in the network dies.

Acknowledging the inherent tradeoff between energy, delay, and delivered traffic, the results of Girici and Ephremides's study confirm that link metric-based policies that jointly considers transmission power requirements, residual energy information, link queue sizes and transceiver utilization provides a balance in terms of energy consumption, average delay and delivered traffic volume. This work is one of the first to address the joint power and link scheduling problem. However, the algorithmic nature and the lack of any interlayer interaction specification limit the approach's applicability to existing architectures. Furthermore, this optimization approach does not specify how to run the required computation in distributed networks, leaving several open issues regarding its implementation in real networks.

9.1.2 Cruz and Santhanam

Cruz and Santhanam [159] present a cross-layer approach with the expanded goals of joint routing, link scheduling, and power control among a network of wireless base stations. Although their approach does not focus on purely ad hoc networks, the cross-layer design concepts in this approach are readily applicable to ad hoc and sensor networks. The main goal for Cruz and Santhanam's approach is to minimize the total average transmission power in the network subject to constraints regarding the minimum average data rate per link and the peak transmission power.

The approach requires synchronization to support its time-slotted structure, and it assumes that the interference conditions do not experience any small time-scale variations. Cruz and Santhanam identify the tradeoff of scheduling wireless links that are far apart to minimize interference and of scheduling links closely to achieve high data rates in the network. As such, Cruz and Santhanam's approach tries to systematically optimize the active subsets of links in the network and their respective transmission powers in order to minimize the total average power expended in the network. The optimal link schedules are also constrained by a minimum data rate and a peak transmission power for each node. The derivation of their optimal policy reveals that each node should either turn off its transceiver or transmit at the maximum allowable transmission power. In order to reduce their algorithm's complexity, they also limit the search space of the optimal schedule by only considering feasible links.

The computational cost for a group of more than 15 links grows quickly, limiting the scalability of this approach. To support larger networks, the authors suggest a hierarchical clustered approach in which each cluster has at most 15 links. The clusters are considered to schedule links independently of each other.

After determining optimal link and power schedules, Cruz and Santhanam's approach takes on the problem of allocating traffic on the scheduled links to support a given traffic matrix describing the required traffic rates of specific source destination pairs. The approach uses a shortest path routing algorithm to incrementally compute the cost of supporting additional traffic on each link, and uses an iterative approach which eventually converges to an optimal route assignment. The optimal routing paths for traffic purposes do not always correspond to minimum energy paths. However, their routing policy uses minimum energy paths whenever the data traffic requirements are low.

At low traffic loads, Cruz and Santhanam's approach reduces to a simple TDMA policy that activates one link at a time in sequence. At higher traffic loads, the approach supports higher data rates than TDMA through concurrent scheduling of links. The cost of higher throughput is reduced energy efficiency, since more nodes and links are involved in the concurrent transmissions. For example, forwarding data along multiple paths may provide higher throughput, but it involves more forwarding activity at intermediate nodes which translates into higher power consumption.

Cruz and Santhanam's approach is general enough to apply to clustered topology ad hoc and sensor networks despite the fact that it develops the optimization approach for a network of wireless base stations. Some of the limiting assumptions of this approach include synchronized node clocks and slow varying interference conditions. Another inherent drawback to this algorithmic approach is the lack of implementation or functional description. It also uses a performance metric of long term rate requirement, meaning optimal solutions may involve significant rate fluctuations.

9.1.3 ElBatt and Ephremides

In a similar approach, ElBatt and Ephremides [160] try to optimize the joint power control and TDMA scheduling problem. The distinction of this approach is that it specifically targets contention-based wireless ad hoc networks, whereas Cruz and Santhanam's approach targets wireless base stations. Elbatt and Ephremides's approach consists of 2 phases:

1. **link scheduling**: The scheduling phase coordinates the transmissions of independent users in order to eliminate strong levels of interference.
2. **power control**: The power control phase runs in a distributed fashion to determine the admissible power vector to be used by scheduled users to satisfy their single-hop transmission requirements.

The mechanisms of this approach attempt to adapt existing TDMA and CDMA cellular network mechanisms for ad hoc networks through cross-layer enhancements. Elbatt and Ephremides develop their model based on several assumptions, of which the following are most notable:

- There is a central controller that runs the scheduling algorithms
- Nodes exchange short control messages through a separate contention-free control channel
- Nodes are aware of the locations of all other nodes in the network
- In addition to interference of neighbor communications, nodes that are further away also cause interference according to a realistic interference model

This approach does not consider mobility or routing aspects. It exploits the notion of spatial channel reuse for nodes that are separated by a distance larger than a computed channel reuse distance.

The approach splits the optimization problems into 2 separate optimization problems. The first optimization problem is the “valid scenario optimization” problem: to find the valid set of nodes, which is the largest subset of network nodes that can simultaneously transmit packets. The second optimization problem is the “admissible scenario” problem, which determines the largest subset of the valid set that can satisfy the SNR constraints of all the active links.

Elbatt and Ephermides propose a distributed power control algorithm modeled after cellular networks. They prove through analysis and simulations that the computation complexity of their algorithm for valid transmissions in both TDMA and TDMA/CDMA wireless ad hoc networks grows exponentially with the number of links. Their simulations also reveal that CDMA systems provide higher single-hop throughput and lower transmission power than TDMA systems, at the cost of higher processing power. Elbatt and Ephermides's cross-layer optimizations also significantly increase the throughput and decrease the transmission power cost over the heuristic case, especially under heavy load conditions.

One advantage of this approach is that it builds on previous cellular network concepts, such as spatial reuse of resources. Unlike previous algorithms that do not address computational distribution, this approach is that it addresses the computation distribution issue, and it theoretically shows that computation can be distributed across the nodes. However, it does not specify how to practically achieve the computational distribution. Multihop routing and mobility are two other issues not addressed in this approach. Finally, the assumption that all nodes are location-aware introduces cost and complexity to this approach.

9.1.4 Kozat

Kozat et al. [161] propose another cross-layer approach that attempts to minimize power consumption in an ad hoc network while providing quality of service guarantees in terms of bandwidth and bit error rate. As in [159] and [160], this approach involves collaboration between the MAC and physical layers. Because Elbatt and Ephermides's approach is inspired by cellular networks, it does not address multihop cases. Kozat's approach extends the single hop considerations of Elbatt's approach to handle multihop networks. Kozat's main distinction from Cruz's approach is that Kozat's approach addresses the short-term throughput, where Cruz's approach focuses on the long-term average throughput. Because obtaining the optimal solution is NP-hard, Kozat et al. propose heuristics for link scheduling and power control with quality-of-service guarantees.

Kozat's approach assumes that there are a number of sessions active through the network, and that each session is predetermined through some routing protocol. Each session consists of a source-destination pair, a set of directed links from source to destination, a short-term end-to-end bit rate, and a per-link bit error rate. This approach assumes that time is divided into slots, and that the scheduling algorithm must assign one or more slots per transmitter during each time frame.

This model also provides a one-to-one mapping between the modulation level, bit error rate and the signal to noise ratio. Because the model assumes that the coding rate and modulation level are kept fixed throughout the frame, it relies on transmit power control and slot assignment for satisfying the bit error rates as the interference level varies with time.

Kozat's approach introduces the concept of virtual links. In certain networks, separate sessions may use the same link for forwarding data. Each instance of the same link that appears on multiple paths is indexed separately and is denoted as a virtual link. As such, several virtual links may correspond to the same physical link.

Although the central aim of this approach is to minimize the total transmit power as summed over all time slots and links while satisfying the minimum rate and SINR constraints of the sessions, Kozat et al. analytically reduce this problem to a classical power control problem in cellular networks once a particular slot is allocated. As a result, they propose two algorithms for solving these optimization problems.

The first algorithm adopts a top-down approach. The algorithm begins with a set of empty time slots in a frame and a set of unscheduled virtual links. The algorithm starts with the first time slot and attempts to fill that slot with as many virtual links as possible. Because this approach assumes half duplex point-to-point communication, virtual links scheduled during the same slot must not have any transmitters or receivers in common. The algorithm then checks if there is a feasible power allocation for the set of scheduled virtual links. In case the allocation is not feasible, high interference links are pruned from the allocation until the power allocation becomes feasible. The two phases in this algorithm resemble Elbatt and Ephermides's 2 phase solution for the link scheduling and power allocation problem.

The second algorithm adopts a bottom-up approach, by performing the iteration over the unassigned links. This algorithm initially assigns exactly one link to each slot until all slots have exactly one link. It always chooses the link with the maximum interference among all the unassigned links to place in the next empty slot. This ensures that transmitters that are within close range of each other are spread onto different time slots. The second stage of this algorithm adopts a water-filling approach. A link is assigned to a slot if this assignment is feasible and if the slot has an acceptable power allocation after this new link assignment. This algorithm completes when all links have been checked or when it is not possible to place new links in any of the slots.

The performance evaluation of this approach through simulations that consider minimum hop routing and minimum power routing reveals that the top-down design strategy (which entails first solving the feasibility problem, then minimizing the power consumption) performs better in terms of the objective cost function. The second algorithm which adopts a bottom-up approach (water-filling) performs better than the top-down design strategy in finding a feasible solution for link scheduling and power allocations.

Regarding the interaction with the routing strategy, Kozat et al. note that including power considerations at the routing layer plays a dominant role in reducing power consumption, but it also may adversely affect QoS guarantees at the data link and physical layers. As in previous approaches, Kozat's approach lacks mobility support, because it considers that link gains remain constant within a time frame.

This approach adopts an open view towards including more layer functionalities, although it focuses on scheduling and power control at the data link and physical layers respectively. This framework's support for integration of higher layer functionalities, such as QoS metrics, contributes to its flexibility. Other strengths of this approach include the realistic assumption of short-term link quality variations, and the introduction of the concept of virtual links. The algorithmic nature of this approach is a limiting factor towards its implementation, especially the centralized computation structure that considers that the input data for the algorithm can be compiled at one location. Finally, this approach optimizes link schedules and transmission powers based on the assumption that routes are already established. In practical scenarios, this assumption is simplistic since routing issues are closely intertwined with link scheduling and power control issues.

9.1.5 Lu and Krishnamachari

Lu and Krishnamachari [162] attempt to extend earlier work on the joint scheduling and power control problem by proposing a tunable framework that manages the tradeoffs between throughput, energy and latency. This approach notes that previous works on joint scheduling and power control [160] attempt to pack the maximum number of links that can be active simultaneously in each time slot. Although the power control phase minimizes the transmission powers on the scheduled links, this scheduling policy does not take energy into consideration and thus may not be energy efficient. Through parametrization of the performance aspects, Lu's approach achieves significant energy savings without sufficiently sacrificing throughput.

Lu and Krishnamachari's approach makes the following assumptions:

- It operates in a static sensor network where all nodes have omnidirectional antennas
- It considers a periodic data gathering application in which nodes send their packets to a sink or to neighbors with a delay bound
- It supports applications with per-hop or end-to-end deadlines. In the latter case, the approach computes per-hop deadlines by dividing end-to-end deadlines by the number of hops along a path
- It assumes a TDMA multiple access strategy
- Each node generates a random amount of packets during a time frame
- It considers the packets to be forwarded at a node as local transmission requests

Lu and Krishnamachari's approach also exploits a more realistic SNR model. Within this model, links can be scheduled concurrently in the same time slot if their SNR values are larger than a certain threshold. The SNR is computed as the ratio of the received signal power to the sum of the noise and the interference of all other concurrent links.

To promote flexibility, Lu and Krishnamachari's approach defines a tunable gain function. The total gain in the network is the weighted difference of the throughput and the energy cost. To simplify the analysis, the weight of the throughput can be fixed to 1 in order to focus the optimization on choosing the weight of the energy cost to serve different applications. For applications that favor energy efficiency, the energy cost has a weight larger than 1. In contrast, applications with high throughput requirement can set a weight for the energy cost that is smaller than 1. The optimization problem is subject to worst case per-hop delay constraints.

When the energy cost weight is set to zero, Lu and Krishnamachari's problem reduces to the same problem in [160]. For the more general optimization problem, several heuristics are proposed.

The first greedy algorithm assumes that the collection of all feasible transmission sets S is given, and this algorithm selects the appropriate link sets to be scheduled in the same time slot by iteratively choosing the next set that maximizes the total gain. The computational complexity of this algorithm grows exponentially with the number of links, making it suitable for small clusters in large sensor networks.

The second heuristic introduces two new assumptions: (1) that link gains change slowly during a frame; and (2) that all nodes can stay informed of all link gains. It also assumes that nodes exchange short control messages at the beginning of every time frame to announce their links to be scheduled. This algorithm then obtains all the feasible transmission scenarios for a time frame and compares the overall gain of each scenario, through the gain function that includes delay, throughput and energy to choose the scenario with the largest gain. The computational complexity of the second heuristic also grows exponentially with the number of links.

The performance evaluation of this approach yields two main results. First, by relaxing the latency bound, the approach can achieve significant energy savings. Second, by varying the weight of the energy cost, the algorithm is able to save significant energy without hurting throughput.

The main advantage of this approach is the provision of a tunable framework for customizing network behavior to different application requirements including throughput, delay, and energy. The assumptions for algorithm development are realistic and not limiting. The drawback of this approach is its strictly algorithmic nature and its lack of implementation details.

9.1.6 Madan

Madan et al. [163] propose another sensor network cross-layer optimization approach for sensor networks that aims at maximizing the network lifetime, defined as the time at which the first node dies in the network. In contrast to prior approaches that target energy efficiency, Madan's approach specifically addresses the key issue of network lifetime in sensor networks. Optimizations for energy efficient behavior do not necessarily yield the longest network lifetime. For example, a strategy that minimizes overall power consumption may not balance traffic load, causing certain nodes to die earlier than others. Madan's approach adjusts transmission rate, transmission power, link schedules, and routes in order to satisfy network lifetime constraints.

The approach advocates multihop routing and load balancing at the routing layer. Multihop routing aims at replacing long communication links with multiple short links to reduce transmission power consumption. Load balancing aims at shifting traffic away from hot spots in the network for more balanced energy consumption. As in the model in [160], Madan's approach also exploits spatial channel reuse for link scheduling.

As in most models discussed in this chapter, Madan's approach targets TDMA sensor networks. The approach also considers that each node generates data at a fixed rate, which needs to be communicated to a single sink node. Because the lifetime optimization problem is NP-hard, the problem can be recast as a convex optimization problem by restricting links to TDMA schedules and by allowing variable length slots.

Madan's optimization algorithm starts with a feasible suboptimal schedule. It computes the optimal transmission rates and powers for this schedule. The algorithm then checks if the schedule is feasible, in which case it disables links with SNR close to 1. It then allocates an additional time slot to a link with the maximum average power. This step repeats until it converges to the optimal solution. At every iteration, the algorithm terminates in case of a repeated schedule, unfeasible schedule, or unfeasible SNR. Checking these conditions ensures that the schedule at the end of each iteration is feasible.

To support partially distributed computation, Madan et. al reformulate the problem to address the optimization of transmission powers and rates for a *fixed* link schedule. In particular, they show that each node can compute a subproblem of the complete optimization problem during each iteration. Of course, distributed computation for this optimization must manage control message exchanges for sharing schedule information, an issue which is not addressed in Madan's approach. Anther open issue for this approach is the realistic definition of network lifetime. Modeling sensor network lifetime requires a comprehensive approach that takes into account power consumption due to sensing, processing, communication, and switching hardware components between different power states.

The main advantage of this approach is its potential for distributed computation, although the mechanisms for distributed implementation are not clear. This approach's performance goal to maximize sensor network lifetime is also a highly practical issue in sensor networks. The drawbacks of this approach include the purely algorithmic nature and the unrealistic definition of network lifetime.

9.1.7 Cui

The same authors of Madan's approach propose Cui's cross-layer optimization approach [164] that targets energy efficiency in *small* scale sensor networks. What distinguishes Cui's approach from previous approaches is that it considers both the *communication power* and the *hardware power* as the causes of power consumption in small scale sensor networks, partially addressing the unrealistic power modeling of their previous approach. The argument is that the hardware power consumption, which includes processing and sensing power consumption, may equal or even overweigh communication power consumption when communication links span short distances. Thus, Cui's approach considers both causes of power consumption to jointly determine communication paths, link schedules, M-ary Quadrature Amplitude Modulation (MQAM) modulation rates and transmit powers that yield better energy efficiency in the network.

This model assumes that the nodes maintain synchronization and operate with a predetermined TDMA schedule. It also assumes that nodes are stationary or quasi-stationary, meaning that nodes may move at very slow speeds. Furthermore, it considers that link schedules are interference-free and periodic; once a node schedules its transmission during a particular time slot in a time frame, it uses the same time slot for its transmissions in subsequent time frames.

Cui's approach addresses a common sensor network topology, where all nodes send their data to a single sink node. In general, this approach requires that each node send its data toward a single destination. As in many periodic data gathering sensor network applications, data becomes available for transmission periodically at each node.

Cui et al. use an energy model that considers two power states: active, and sleep. In the active mode, the node is either transmitting or receiving data. During sleep periods, the node shuts off all of its circuitry. The power consumption due to leakage current and the transitional phase for switching between active and sleep mode is neglected. This approach uses the circuit model from [165] for the transmit and receive signal paths. It also neglects the energy consumption of baseband signal processing blocks, such as for source coding, pulse-shaping, and digital modulation.

Finally, this approach explores energy-delay tradeoffs related to this cross-layer optimization problem. In particular, Cui et al. consider that queuing delay and transmission delay are the sole contributors to packet delay at a node, thereby ignoring propagation delay which is negligible in small scale sensor networks. A notable observation is that in a TDMA tree topology, scheduling a node's incoming links before its outgoing links on a particular path minimizes packet delay. The validity of this claim can be seen by examining the reverse case, where nodes schedule their outgoing links before their incoming links. A node would send its packets from the last time frame along its outgoing link, meaning the packet for this period would experience a delay equal to the time frame period. Packets would experience this delay at every hop to the destination, yielding a relatively large end-to-end delay in multihop networks. The authors also show how to generalize the tree topology problem to a random graph topology.

One strength of this approach is the joint consideration of routes, link schedules, transmission power, and modulation rates. Another novel contribution is the consideration of circuit power consumption in addition to transmission power consumption. However, this approach neglects other causes of power consumption, such as baseband signal processing blocks, leakage currents and the transitional phase for switching between active and sleep mode. It is also a highly algorithmic approach that does not specify implementation details for mapping the algorithm to interlayer or internode interactions in existing architectures. Finally, the predetermined TDMA schedule in this approach is a limiting factor for sensor networks with dynamic data reporting conditions.

9.1.8 Wang and Kar

Wang and Kar address the rate control problem in multi-hop ad hoc networks in order to achieve proportional fairness among end-to-end sessions [166]. In particular, their approach considers the complex problem of maximizing the bandwidth utilization while maintaining fairness among active sessions. According to Wang and Kar, the dependance of link rates on MAC parameters such as transmission probabilities or back-off window sizes necessitates joint optimization at both the transport and data link layers. The main challenge is to specify the cooperation between the data link layer and the transport layer so that the aggregate utilities of all end-to-end sessions are maximized. To solve the joint optimization problem, Wang and Kar propose two algorithms: a dual-based algorithm, and a primal-based algorithm.

In the dual-based algorithm, active sessions adjust their rates in a distributed manner in order to achieve fair session rates given a set of specific link rates. The algorithm relies on iterative optimization between the link and transport layers, albeit at different time scales. Each node periodically updates the link rates (at the data link layer) using information collected from the local neighborhood on current link prices and link attempt probabilities. Whenever the link layer updates attempt probabilities, the transport layer begins an iterative search for the optimal end-to-end session rates and optimal link prices (on the basis of the updated link rates). Thus, the transport layer operates at a smaller time scale and the link layer operates at a larger time scale.

The primal-based algorithm transforms the non-convex joint optimization problem into a convex one through simple transformations. The main distinction of the primal based algorithm is that the transport and data link layers work at the same time scale. While the link layer updates link rates using the link attempt probabilities of neighboring nodes, the transport layer updates the session rates using aggregate traffic load and capacity information of the links on the active paths. Updating configurations in parallel at both the data link and transport layers yields a faster convergence rate to the optimal solution in comparison to the dual-based algorithm.

Both of the proposed algorithms attempt to optimize a logarithmic utility function of the session rates. The choice of the logarithmic utility function is well-suited for the target problem: the proportionally fair rate control optimization problem. The difference between the two algorithms conveniently illustrates some of the trade-offs of cross-layer design strategies. While both algorithms couple optimization at the transport and data link layers, the dual-based algorithm leans more towards information-sharing, maintaining modularity between the transport and data link layers. Programmatically, the dual-based algorithm converges more slowly to the optimal solution since it involves embedded loops between the data link and transport layers. The transport layer iteratively works in the inner loop to optimize session rates and link prices, while the data link layer works in the outer loop to adjust the link attempt probabilities and link rates. The embedded loop structure causes a longer convergence time. In the primal-based algorithm, both the transport and the data link layer adjust their configuration according to the same time scale. As a result, the inter-layer interaction occurs at each iteration, which enables faster convergence to the optimal configurations. The primal-based algorithm improves convergence time through a closer design-coupling between the transport and data link layers.

The main advantage of Wang and Kar's approach is the distributed nature of both algorithms. In particular, both algorithms rely on link information that can be collected locally at each node to optimize the utility function and to promote fairness. One drawback of this approach is its iterative nature, which risks intensive and ongoing computation in mobile or resource-limited nodes in ad hoc and sensor networks. In the case of highly dynamic or mobile networks, the algorithm may initiate often, sometimes even before it reaches the optimal configuration for the previous epoch. Furthermore, this approach assumes a symmetric hearing matrix, which does not capture potential asymmetries of unidirectional wireless links. Nonetheless, Wang and Kar's approach demonstrates the benefits of cross-layer interaction between the transport and data link layers.

9.1.9 Merz

Merz et al. take a step away from algorithmic design coupling toward a more functional approach in cross-layer design for ad hoc networks [172]. In particular, their approach integrates functions from the physical and MAC layers in UWB ad hoc networks to maximize network throughput. It is worth noting that the move from algorithmic to functional design required addressing a more specific network scenario that uses UWB technology.

Merz's approach considers the spread-spectrum nature of UWB networks, which separates concurrent transmissions on the same channel through time-hopping codes. The price for supporting multiple transmissions is that concurrent transmissions cause mutual interference, which reduces link qualities. Reduced link qualities require either the increase of the transmission power or the reduction of transmission rates. Since Merz's approach aims at always using the maximum transmission power on all links, it proposes the dynamic adjustment of active link rates with the level of interference. The approach achieves rate adjustment through dynamic channel coding. The dynamic channel coding algorithm employs incremental channel redundancy. A source node sends the receiver packets with a high code rate. As long as the receiver cannot decode the sender's packet, the receiver sends negative ACKs (NACK) back to the sender. Upon receiving NACKs, the sender lowers the code rate (increasing redundancy) and tries sending another packet with the lower rate. This process repeats until the sender receives an ACK message from the receiver indicating that it could decode the most recent packet. At that point, data transmission proceeds with the highest possible code rate.

In addition to the handshake for code rate determination, Merz's approach closely couples the MAC layer functionality with the underlying UWB physical layer. The MAC protocol adopts receiver-specific time-hopping codes. A sender always uses the intended receiver's code for transmissions. After sending a data packet, a source node listens on the receiver's TH-code for an ACK message. Once it receives the ACK, the sender listens on its own TH-code again. Merz et al. also specify signalling procedures for indicating idle channels, deferring transmissions, back-off timers, and failed transmissions.

The advantages of this approach include its support for distributed computation and its specification of the relevant node interactions. Another strength is the introduction of interference mitigation as an alternative to mutual exclusion in UWB networks. The main disadvantage of this approach is that it does not support rate guarantees for real-time applications, since the transmission rate is always dependent on interference.

9.2 Information Sharing Approaches

Information sharing approaches preserve the existing layer functionality of a reference architecture while providing novel means for interlayer interaction, in attempt to balance generality and performance. Information sharing approaches generally adopt a more functional and architectural perspective of cross-layer design through consideration of the practical internode and interlayer interactions to support cross-layer optimization algorithms.

Table 9.2 provides an overview of the features of existing cross-layer approaches for ad hoc and sensor networks that rely on information sharing. The columns in Table 9.2 correspond to the features, while the rows correspond to existing approaches. The remainder of this section expands on the classifications in Table 9.2.

9.2.1 Sichitiu

Sichitiu's approach for cross-layer optimization of sensor networks [171] departs from the purely algorithmic design of other approaches with proposals for distributed implementation and integration with existing protocols. This approach proposes a deterministic schedule-based strategy that relies on sleep modes for promoting energy efficiency in sensor networks. It relies on the close coupling of the MAC and network layer for determining optimal schedules.

Sichitiu considers a realistic power consumption model that includes idle listening, control packet overhead, retransmissions, high transmission powers, and suboptimal routes. His approach targets data gathering sensor networks where nodes are stationary and traffic flow is long-lived, periodic, and predictable. Finally, Sichitiu considers that nodes maintain perfect synchronization and that each node has one data flow to be scheduled during a data reporting period.

This approach distinguishes between two phases for each flow in the network:

1. **Steady state phase:** Nodes spend most of their time in steady state phase, during which nodes take action according to a fixed schedule table.
2. **Setup phase:** Node failures, battery depletion, or a change in the network objective may cause a flow to enter a short-lived setup and configuration phase, before going back to the steady-state phase.

The schedule tables at each node consist of three actions: sample, transmit, or receive. The sample action corresponds to sampling a sensor. Similarly, a transmit action corresponds to a packet transmission, and a receive action corresponds to a packet reception. The schedule table specifies the type of actions to be taken at a node and their respective schedule.

The approach notes an inherent property of tree topology sensor networks: nodes closer to the data sink have a larger forwarding load than leaf nodes, so they must keep their transceivers on for a longer time. Sichitiu also observes that the transceiver wake-up power is not negligible, which means that compact schedules are preferable to schedules that require a node to wake up several times during one time period. In case more than one schedule are equivalent for power purposes, this approach chooses routes based on compactness, delay, or load balancing considerations. A major issue within this approach is the operation during the setup and reconfiguration phase, which is split into 2 steps: route selection, and route setup. The underlying routing protocol, which is left unspecified in this approach, handles the route selection step. The route setup step involves sending probe messages that can find appropriate schedules for the data transmission and reception on the links of the selected path. The route setup probes use a generic RTS/CTS MAC layer.

Model	Goals			Target	Input Aspects					Optimizations				Design
	1	2	3	Network	App.	Mid.	Net.	D.L.	Phy.	Mid.	Net.	D.L.	Phy.	
Sichitiu [171]	network lifetime			time driven sensor networks	generic app. var.		route hops	link quality			route costs	link schedule		distr.
Chen [167]	data access			multi-media ad hoc networks		priority	location/ mobility info			data compressn replication	routes			distr.
SP [168]	efficiency	generality		sensor networks			packet reliability /urgency	link active schedules /costs			route flows	link schedule		distr.
Jurdak [100]	energy efficiency	network life		sensor networks	sens. behav.		route flow	radio duty cycle	sens. duty cycle		route costs	LPL mode		distr.
Jurdak [100]	maximum through-put	fair access	minimum latency	UWB ad hoc networks	fail. notif.		traffic bandwidth		SNR		route costs	hello message period	trans. rate/ power	distr.
Jurdak [100]	network life	energy efficiency		acoustic sensor networks	report. freq.		route flow		trans. freq./ dist.				trans. freq.	unsp.

Table 9.2. Applied Cross-Layer Approaches based on Information Sharing

The route probe message schedules transmit and receive actions at each intermediate node along the path, until it reaches the data sink. The intermediate nodes store the schedule in a temporary table, until they receive an acknowledgement of the schedule from the data sink. The acknowledgement takes the reverse path to reach the original sender.

In order to ensure that control packets do not interfere with data flow, Sichitiu proposes a two-priority approach. Data packets are sent immediately when scheduled, while control packets are sent after waiting for a short duration. Another issue is that nodes should be awake to receive the routing probe messages, which presents a significant overhead for this scheme. Additionally, nodes must store 2 entries for each flow, corresponding to the transmit time and the receive time. In terms of scalability, the overhead communication of this approach increases exponentially with the number of nodes causing the nodes close to the sink to remain awake most of the time, shortening network lifetime.

The functional nature of this approach that describes internode interactions and intranode interactions is a definite strength. Another advantage is the realistic power consumption model for sensor nodes, which takes into account the transceiver wake-up power.

The drawbacks of this approach include its limited applicability, since it targets monitoring sensor networks with long-lived, periodic, and predictable traffic flow. This approach also has limited scalability because of the communication overhead for maintaining synchronization and the storage overhead for maintaining active path information.

The two other information sharing approaches considered in this section apply cross-layer architectural models from Chapter 8 to specific scenarios. These approaches aim to bridge the gap between the algorithmic and functional optimization on one hand, and the architectural aspects on the other.

9.2.2 Chen

Chen [167] proposes a cross-layer optimization approach for multimedia applications running over ad hoc networks that involves information sharing between the routing and middleware layers. This approach aims to provide end-to-end QoS guarantees in resource-limited mobile ad hoc networks. Figure 9.1 shows the approach's cross-layer interaction.

In Chen's model, the application layer generates and shares multimedia data with other users in the network. The middleware layer is responsible for locating, accessing, and replicating data to applications through the data accessibility service. The network layer computes feasible routes and forwards packets to other mobile nodes in the network. The framework includes a system profiles component through which the routing layer and middleware layer share information. The system profiles component is essentially similar to the Network Status component of MobileMan.

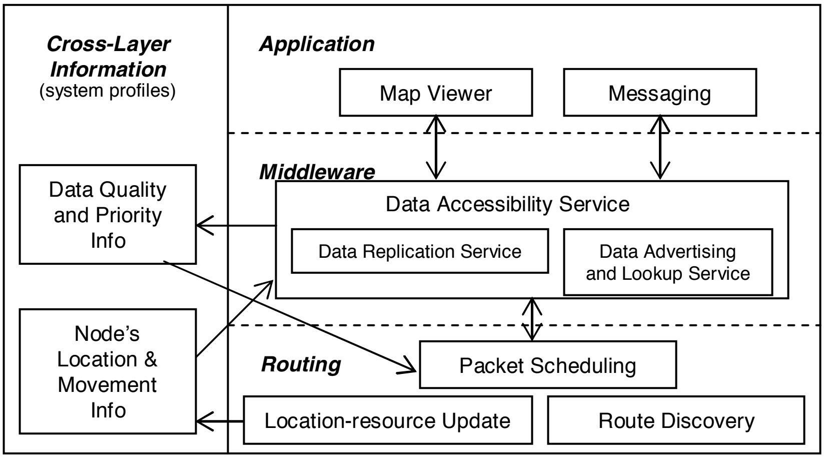


Fig. 9.1. Chen's cross-layer interaction model [167] (©2002 Springer*)

At the network layer, this approach adopts a predictive and proactive location-based routing protocol. Nodes periodically declare their location information, through which other nodes can learn the location and mobility patterns. This location-awareness enables nodes to maintain and update routes with specific sets of QoS parameters. The routing protocol in this model requires each node to the following state information: a data update table, and routing state tables.

The middleware layer has two tasks:

1. To obtain and present data availability information to the application level. In this step, the QoS parameter of interest is the success rate in accessing data.
2. To retrieve the data from a remote host with certain application-level requirements, such as data access deadline and data quality. The middleware layer can translate the application-layer performance requirements into network layer QoS parameters such as reliability, bandwidth or delay, enabling the network layer to set up a route according to these parameters.

The system profiles component includes node location information and mobility patterns from the network layer, while the middle layer provides data priority information. Both layers use each other's state information to drive optimizations. The middleware layer utilizes the node location and movement pattern to predict future connectivity of a group in order to determine data replication policies. The network layer utilizes the data priority information to differentiate and prioritize network level packets for scheduling purposes.

In addition to information sharing through the system profiles, the network and the middleware layers actively communicate with each other via signal-

---

*Reprinted with kind permission from Springer Science and Business Media ing regarding the condition of data transmission and adaptation. To adapt to minor network changes, the network layer can modify the current route, recompute another route, or notify the middleware layer of QoS violations in case no other route is available. In the latter case, the routing layer provides hints of currently available routes and their characteristics such as bandwidth, delay, and stability to the middleware layer. With these hints, the middleware layer can determine a proper higher-level adaptation strategy such as data format compression.

The cross-layer interaction and adaptivity of this approach perform well for small networks only. To achieve scalability, Chen et al. suggest flooding state updates locally in a hierarchical topology, as suggested in [159].

In sum, Chen's applied cross-layer approach confirms that information sharing between the middleware and routing layers enable applications to achieve higher success rates in accessing data even in a highly dynamic ad hoc networks.

9.2.3 Sensor Protocol

Polastre et al. propose an implementation of the Sensor Protocol (SP) [168] within the SP architecture (see Chapter 8). The proposed SP protocol implementation attempts to provide a unifying abstraction that balances generality and efficiency in sensor network design. In particular, it introduces interfaces for efficient cross-layer information sharing between the MAC and routing layers. The interfaces are designed to provide interoperability with a wide range of MAC and routing protocols, promoting innovation through code reuse.

The SP protocol implementation provides for three functionalities: data transmission, data reception, and neighbor management. Whereas the first two functionalities are self-explanatory, neighbor management enables the routing and MAC layers to cooperate in managing *internode* interaction.

The *interlayer* interaction between the MAC and routing layers relies on two data structures: neighbor table, and message pool. The neighbor table is a cross-layer entity that enables the MAC and routing layers to store and share neighborhood information, such as link states, route costs, and neighbor sleep and active schedules. The neighbor table thus serves as a common repository for cross-layer state information, saving both storage space and access time to state information. The second data structure in SP, the message pool, enables network protocols to request message transmissions on the basis of higher layer quality requirements. Rather than storing actual messages, the message pool contains references to messages, which can be accessed either in-order or out-of-order, depending on quality-of-service requirements. The network and MAC layers use the message pool structure to pass message information to each other. For example, the network layer can set an *urgent* or a *reliability* bit for a particular message to force the MAC layer to aggressively attempt to meet the delay or reliability requirements for that message. The link layer can also send feedback to the network layer through the message pool. For instance, the link layer can communicate congestion or phase conflicts to the network layer so that the latter can adjust its behavior accordingly.

A significant aspect of the SP protocol implementation is its focus on programmatic issues, such as code size, code reuse, and interlayer interfaces. The proposed SP protocol provides new interfaces for information sharing between generic MAC and network layers within the TinyOS [191] operating system.

The main goals of this approach is to provide architectural modularity and interoperability while maintaining efficient network behavior for a wide set of network and data link protocols. To evaluate SP's success in achieving those goals, the authors perform experiments that couple SP with two MAC protocols of distinct types: BMAC [56] (a channel sampling protocol), and IEEE 802.15.4 [189] (a slotted protocol). The experiments reveal that the SP abstraction layer provides modularity without any significant impact on data delivery for both MAC protocol types. The SP protocol also improves the adaptivity of the BMAC protocol, enabling nodes to adjust more quickly to surges in throughput requirements. Furthermore, deployment experiments illustrate that the introduction of the SP abstraction layer reduces the average duty cycle of nodes for BMAC by more than 50%, through more adaptive duty cycles[1]. The authors also consider three different routing protocols (Mintroute [197], Trickle [169], and Synopsis Diffusion [170]) in their performance evaluation to verify that SP supports various network protocol types above it, in addition to its support for several MAC protocol types below it.

The benefits of the SP protocol include its long-term vision for promoting innovation through a seamless cross-layer architecture. Another added value of this approach is the consideration of implementation details, including data structures for cross-layer information sharing, programmatic issues, and storage complexity. A drawback of the proposed SP protocol is that it leaves several issues open, such as the sharing of timestamping, naming, and security information across layers. To incorporate the additional cross-layer information, SP may require creating new data structures or extending the proposed data structures, which may cause the current SP memory footprint to overgrow existing storage resources at each node.

9.2.4 Jurdak

Jurdak [100] provides another example of a functional/architectural approach in the form of a general cross-layer framework for ad hoc and sensor networks that provides both modularity and flexibility. The framework defines a general structure for implementing optimizations, and different applications can customize the frame work according to their performance requirements.

---

[1] Chapter 10 elaborates more on adaptive duty cycles with BMAC, as proposed independently by Jurdak et al. [193].

The modular design of Jurdak's framework enables the use of any optimization algorithm to modify node configurations. The flexible node state definition enables customization for different applications. By choosing the configurations on which the algorithm operates, the framework can support different applications, quality goals, and communication technologies. The remainder of this section briefly demonstrates how customizing Jurdak's general framework to particular situations yields significant performance benefits. In particular, the framework is customized to three scenarios: monitoring and event-driven RF sensor network test-bed; UWB ad hoc network; and acoustic underwater sensor network. Chapters 10, 11 and 12 discuss these 3 case in much more detail.

RF Sensor Network

The first scenario for this framework is a sensor network test-bed that uses RF waves [100]. The network targets long-term monitoring deployments in which the sensor nodes periodically sample their sensors and send the data towards a single base station. Because of the long-term operation requirement, the goal in this network is to promote energy efficiency through adaptive listening modes and to provide a longer network lifetime through load balancing.

To achieve these performance goals, this scenario relies on allowing sensor nodes to power down their radio when they are inactive. In particular, nodes use BMAC [56] to periodically check the channel for activity. The single data sink in the network yields a tree topology, placing more forwarding load at the nodes close to the base station. As such, Jurdak's approach introduces Adaptive Low Power Listening (ALPL), an algorithm which runs locally at each node to enable the node to adapt their check interval according to their current state. Unlike most other optimization algorithms, ALPL also specifies the functional and implementation details including signalling, packet formats, and interfacing with existing mechanisms within the TinyOS [191] environment.

Referring back to Fig. 8.6, Jurdak's framework consolidates three aspects into the node state for this scenario: (1) number of descendants in the routing tree; (2) radio duty cycle; and (3) node roles. The number of descendants aspect quantifies the forwarding load of each node. The duty cycle aspect determines how busy a node's radio has been during the recent time window. Finally, the role aspect corresponds to the sensing and processing power consumption that may contribute to load imbalance in the network.

After collecting neighborhood state information, ALPL locally runs on each node with the node's overall state information as an input. Subsequently, ALPL locally adjusts the MAC layer listening mode and the routing cost of neighbors. Performance evaluation on a deployed sensor network test-bed has shown that Jurdak's framework yields significant global energy savings and it effectively balances the load in this scenario. Chapter 10 elaborates further on this case study.

UWB Ad Hoc Network

Jurdak's framework is also customized for an UWB ad hoc network, resembling the network in [172]. This scenario targets QoS provision in UWB ad hoc networks. The performance goals of this scenario are four-fold: (1) maximize throughput; (2) promote fairness; (3) minimize link setup latency; and (4) minimize control overhead.

The spread-spectrum properties of UWB discussed in Section 9.1.9 imply that neighboring links can coexist, but they cause mutual interference. To fully exploit these dependencies, this case study customizes Jurdak's framework to provide a joint routing/MAC approach based on a broad node state specification. The local state at each node combines interference, received radio power, communication resource allocation for different traffic classes, and node reliability. Nodes use this state information to adapt the transmission rate and power of their active links in order to maintain quality guarantees. Nodes also adapt the neighbor costs and the period of their periodic hello messages according to changes in local and neighborhood state.

Within the customized optimization framework, this scenario uses U-MAC, an adaptive MAC protocol for UWB networks in which nodes periodically declare their current state, so that neighbors can proactively assign power and rate values for new links locally in order to optimize global network performance. Simulations comparing U-MAC to the reactive link setup approach confirm that U-MAC lowers link setup latency and control overhead, doubles the throughput and adapts better to high network loads. For a more detailed description on this case study, please refer to Chapter 11.

Acoustic Underwater Sensor Network

The final scenario customizes Jurdak's framework for an acoustic underwater sensor network. This scenario considers a network architecture in which sensor nodes are deployed underwater to sample the environment. A single surface node acts as the gateway between the underwater nodes and the network operators. The underwater nodes communicate acoustically with each other and with the gateway node, which is equipped with long range radio that can send the data to a central repository on land. Thus, the logical topology of this network is also a tree topology centered at the gateway node.

The specific application of interest is the deployment of an underwater acoustic sensor network for environmental monitoring. The main challenge of deploying such a network is the limited battery resources of individual sensor nodes, which requires frequent retrieval and recharging of the node. Thus, the performance goal in this scenario is the maximization of network lifetime and the minimization of power consumption. This case study applies Jurdak's optimization framework to an underwater acoustic sensor network for monitoring of a watershed in order to achieve the performance goal through distributed local decisions.

The focus of the case study is the development of the optimization algorithm that maximizes battery life and minimizes energy consumption for an underwater sensor network. The state parameters that affect energy consumption are: (1) internode distance; (2) transmission frequency; (3) frequency of data updates; and (4) number of descendants in the routing tree. Transmission frequency and internode distances are both physical layer aspects. The frequency of data updates is an application layer aspect, and the number of descendants in the routing tree is a network layer aspect. The tree topology of this network causes nodes close to the gateway to deplete their energy resources quickly. To address this issue, the algorithm enables the loaded nodes to select low transmission frequencies for their communication in order to reduce their power consumption. The rationale is that signal attenuation underwater depends on both distance and frequency. By assigning lower transmission frequencies to loaded nodes and higher transmission frequencies to unloaded nodes, the algorithm promotes load balancing and a prolonged network lifetime. The local and neighborhood state information can be employed locally at each node to dynamically select the appropriate transmission frequency. Chapter 12 presents the details of this scenario.

9.3 Global Performance Goals

This section discusses the common performance goals of cross-layer approaches for ad hoc and sensor networks. In other words, the focus of this section is to address the question: What performance goal is this particular cross-layer approach trying to achieve? Most approaches set performance goals on a network-wide basis to provide system-wide QoS guarantees. Some of the approaches address tradeoffs between several performance goals, while others focus on a single performance goal. Each subsection in this section focuses on one performance goal, with a discussion of possible interactions with the other quality metrics.

9.3.1 Maximize Network Lifetime

One of the most common performance goals for ad hoc and sensor networks is to maximize network lifetime. Because resource-limited nodes must handle all aspects of network management in a distributed manner, all internode and interlayer interactions in the network have a high impact on network lifetime.

For example, Girici's approach [158] attempts to prolong network lifetime for ad hoc networks by including the residual battery energy at each node into the routing metric. This approach notes a significant increase in the network lifetime (measured as the time of first node death) as a result of the inclusion of the residual energy metric. The inclusion of energy considerations into routing also yields a minor improvement of cumulative throughput at the time the first node dies.

Interestingly, the remainder of cross-layer approaches that aim to prolong network lifetime all target sensor networks. For instance, Sichitiu's approach [171] considers a realistic power consumption model for sensor networks that incorporates radio power consumption for idle listening, transmission and reception, sleep modes, and switching between modes. Within this approach, the routing and MAC layers can use the power consumption information for optimizing their decisions in favor of network longevity. Sichitiu notes a significant degradation in network lifetime as a result of increased node density.

Madan et al. [163] also address the network lifetime performance goals through cross-layer optimizations. Their approach considers the joint enforcement of mechanisms from the routing, MAC and physical layer such as multihop routing, load balancing, and interference mitigation to prolong network lifetime.

Finally, Jurdak [100] proposes an algorithm for prolonging the lifetime of an underwater acoustic sensor network. Their algorithm analyzes underwater acoustic communication processes to develop a direct relationship between power consumption for data transmission and reception on one hand, and network battery lifetime on the other. The algorithm attempts to favor highly loaded nodes near a base station through smart frequency channel assignments.

9.3.2 Energy Efficiency

The overwhelming majority of cross-layer approaches set energy efficiency as one of their performance targets [100,158–162,164]. Although energy efficiency and lifetime maximization are closely correlated, the correlation is not always positive. Consider a tree topology sensor network, as shown in Fig. 9.2. Most of the routes that minimize energy consumption in the network may traverse the black node near the base station (Fig. 9.2(a)). Because this node has to forward a large number of packets, it depletes its energy resources much sooner than other nodes. In contrast, the routing graph in Fig. 9.2(b) optimizes network lifetime by balancing the network load among the three nodes near the base station. This example demonstrates that the minimum energy routing strategy does not necessarily correspond to the longest network lifetime strategy.

The approach of Girici and Ephremides [158] addresses both issues of network lifetime and energy efficiency. They define network lifetime by the time of the first node death. Their measure for energy efficiency is the energy expenditure per packet. Their findings reveal that including residual energy into the routing metric prolongs network lifetime but it also increases overall power expenditure. In some cases, inclusion of the residual energy into routing decisions replaces minimum energy paths with maximum lifetime paths, trading off overall power consumption for network longevity.

Cruz and Santhanam [159] also target energy efficiency through another perspective on the tradeoff between energy efficiency and throughput in their

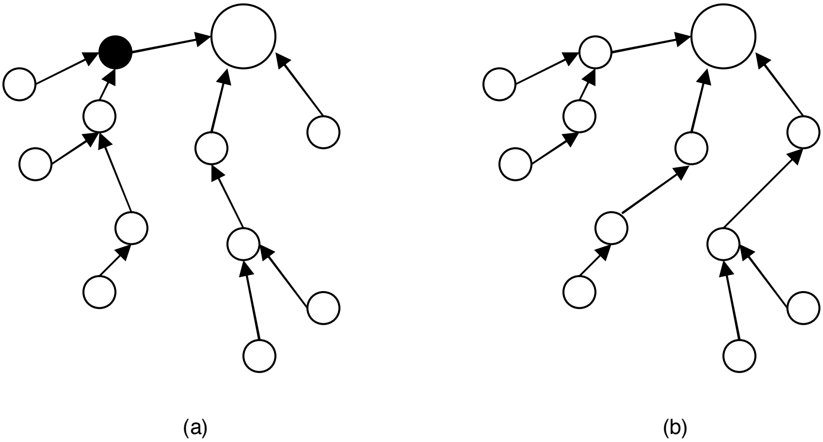


Fig. 9.2. (a) Optimal energy-efficient routing graph. (b) Optimal lifetime routing graph.

cross-layer approach for wireless base stations. Using multiple paths for routing data from source to destination shifts forwarding load away from hot spot nodes and increases throughput. However, multipath routing also causes higher power consumption since it involves more intermediate nodes that must forward data.

Other approaches focus on energy efficiency independently of lifetime considerations. For example, Elbatt and Ephermides [160] focus on energy-efficient performance through a distributed power control algorithm that yields the vector of minimum transmission powers for a given link schedule. Kozat et al. [161] approach the same energy efficiency problem, but they try to generalize Elbatt and Ephermides's 1-hop model to a multihop model. Considering both a top-down and bottom-up approach for determining the link schedule, Kozat et al. determine that the top-down approach of first solving for a feasible link schedule then determining transmission power assignments yield the highest performance. They also highlight the tradeoff between energy efficiency and other QoS metrics, such as delay and throughput.

Lu and Krishnamachari's [162] approach attempts to formalize these tradeoffs between energy efficiency, delay, and throughput through a tunable framework. In particular, the approach notes that relaxing the delay bounds in the network could significantly improve energy efficiency. Determining the optimal weights of the three performance metrics can also lead to power savings with no adverse effect on throughput.

Cui et al. [164] also address the energy delay tradeoff. Unlike previous approaches, their approach incorporates both communication and hardware power consumption. This approach uses sleep modes and smart scheduling of wake-up times for achieving energy efficiency.

Jurdak's [100] approach for RF sensor networks also uses sleep modes in the form of low power listening modes to achieve energy efficiency. Unlike schedule-based approaches, Jurdak's approach requires no synchronization as the underlying BMAC protocol matches nodes' check interval and preamble length to ensure delivery in asynchronous scenarios. In addition to this approach's explicit energy efficiency goal, it also implicitly prolongs network lifetime through load balancing.

Jurdak's [100] applied framework to underwater sensor networks also jointly addresses network lifetime and energy efficiency. Because the focus of this approach is to maximize the network lifetime, the lifetime maximization approach *improves* energy efficiency over the random case but it does not guarantee the highest energy efficiency.

9.3.3 Maximize Throughput

The main purpose of any network is data delivery. Ad hoc and sensor networks are no exception, which is why maximizing the data throughput is a common performance goal of five of the cross-layer approaches for ad hoc and sensor networks in this chapter [100, 160, 162, 166, 172]. Interestingly, all but one of the approaches that optimize throughput rely on *time synchronization* and a slotted time structure. Furthermore, three of the five approaches [100,160,172] adopt a spread spectrum technology such as CDMA or UWB, highlighting the high bandwidth potential of these technologies.

For instance, Elbatt and Ephermides [160] emphasize the single hop throughput in their approach. In particular, their approach optimizes the throughput of individual outgoing time slots for a particular node. A joint CDMA/TDMA scheme provides a higher throughput than a pure TDMA scheme because it enables multiple concurrent transmissions on the same time slot. The tradeoff of using CDMA is the added computational and hardware complexity for transceivers.

Within their tunable framework for TDMA sensor networks, Lu and Krishnamachari [162] define throughput as the number of packets that can be transmitted within a time frame. Because their scheme relies on TDMA, it can only schedule concurrent transmissions that are spatially separated by some distance. As such, the achievable throughput of this scheme is lower than the CDMA/TDMA scheme of ElBatt and Ephermides.

An alternate approach is the maximization of the bandwidth utilization in the network while maintaining fairness among active sessions, as proposed in [166]. Unlike the other surveyed approaches that maximize throughput, Wang and Kar's approach does not adopt a time-slotted structure. However, their approach does have a locally synchronous flavor represented by the periodic update of link attempt probabilities. Another distinction of Wang and Kar's approach is its maximization of end-to-end throughput rather than the local node throughput.

Both cross-layer approaches for UWB also target throughput optimization. Merz's approach relies on interference mitigation for supporting a large number of concurrent UWB links. The use of code rate adaptation also ensures that the nodes use the highest possible transmission rate supported by the current interference levels. Jurdak's approach for UWB ad hoc networks also tries to maximize throughput through adaptation to interference levels in the network, but it does so through power control rather than rate adaptation. This approach considers reserved bandwidth traffic that requires rate guarantees in addition to best-effort traffic. The rate guarantees in the reserved bandwidth case do not allow rate adaptation, so power control is used to offer throughput guarantees.

9.3.4 Minimize Delay

Delay is an important performance metric for time-sensitive applications, such as multimedia delivery or emergency response. Cross-layer approaches typically address delay as a tunable performance goal that involves tradeoffs with energy and data delivery. Thus, delay has not been the *primary* performance goal of cross-layer approaches.

For instance, Girici and Ephremides [158] represent link delay through a routing metric that quantifies the queue size at the sender. This delay routing metric is part of a larger weighted routing cost function that enables different applications to set the metric weights on the basis of their performance requirements. Lu and Krishnamachari [162] present a similar tunable framework that includes adjustable delay, energy, and throughput components. To illustrate the delay energy tradeoff, this approach conducts simulations that relax the delay bounds to increase energy efficiency.

Finally, Jurdak's approach for an UWB ad hoc network [100] explicitly addresses a single delay component: link setup delay. The approach outlines mechanisms for reducing link setup latency through a reduction of handshaking requirements prior to setting up a new link. This approach implicitly attempts to reduce queueing delay by enabling more concurrent links, thereby reducing the probability of deferred transmissions.

9.3.5 Promote Fairness

Optimizing certain performance metrics in the network may lead to unbalanced resource allocation. For example, a greedy algorithm that enables a node to grab the largest available bandwidth may lead to bandwidth starvation of other nodes. Another example is minimum energy paths, which eventually leads to a shorter network lifetime because of overworked nodes on these paths.

Wang and Kar [166] subject their utility maximization approach to a proportional fair rate constraint. They attempt to balance bandwidth allocations among the active end-to-end sessions while maximizing bandwidth utilization.

Another property that contributes to fairness in Wang and Kar's approach is that it evenly spreads bottleneck links in the network, which prevents bandwidth starvation of particular sessions due to overloaded links.

Jurdak's approach [100] also addresses the fairness issue in the context of an UWB ad hoc network. Because UWB is a spread spectrum technology, it has a soft capacity: concurrent transmissions can take place, but the transmission rate of one link is highly sensitive to other neighboring links. At network startup, no links are active so the first node that has data to transmit establishes a link with a high transmission rate, corresponding to a high transmit power (in order to maintain SNR at the receiver). Subsequent links in the neighborhood can only use low transmission power, so their bandwidth is severely restricted. Additionally, UWB suffers from near/far effects evident in other spread-spectrum technologies [17]. Nodes closer to a receiver typically grab a larger portion of bandwidth than nodes further away.

To mitigate these effects, Jurdak's approach sets tunable margins on the portion of bandwidth and emitted power that each node can reserve. The margins yield fairer access to the medium among near and far nodes. It also ensures fairer transmission rates for link requests that arrive at different points in time.

9.3.6 Data Accessibility

From the perspective of multimedia applications running over mobile ad hoc networks, *data accessability* is crucial for ensuring timely delivery and playback of multimedia streams. Providing application with access to multimedia data often requires data replication across several nodes, so that if one node moves away or leaves the network, data forwarding and delivery can proceed unaffected. As discussed in Section 5.2, the middleware layer is responsible for data replication. However, the degree and locality of data replication in ad hoc networks is highly dependent on the nodes' mobility and locations [167], which are both routing layer aspects. Sharing information between the routing and middleware layers can lead to more efficient data replication, providing a higher degree of data accessability and availability to applications.

9.3.7 Efficiency and Generality

Cross-layer proposals typically trade off generality and efficiency. For instance, design coupling approaches aim for optimizing behavior with little regard for interoperability, architectural issues, or code reuse. The SP protocol [168] attempts to provide both efficiency and generality through an implemented information sharing approach. More specifically, it proposes and implements an abstraction layer protocol for sensor networks, with a role similar to that IP within the Internet. The SP design tries to share cross-layer information between the MAC and network layers in a timely and reliable fashion, while keeping in mind the resource scarcity of sensor nodes. To balance generality and efficiency, SP provides cross-layer interfaces, such as for data transmission and feedback, and cross-layer data structures, such as neighbor table and message pool, that are commonly employed by virtually all sensor network and MAC protocols, emphasizing the generality aspect. Each protocol can customize its use of the common interfaces and message structures to achieve efficient behavior.

9.4 Target Networks

Cross-layer approaches are often designed with particular network scenarios in mind. From example, many approaches adopt a slotted time model and assume that nodes can synchronize their clocks. The optimization algorithm itself builds on the target network model and the underlying assumptions. As such, understanding the benefits of each applied cross-layer approach partially relies on considering its network target scenario. This section classifies and discusses applied approaches based on their target network choice. The cross-layer approaches target 2 broad classes of networks: ad hoc networks, and sensor networks.

9.4.1 Ad Hoc Networks

Cross-layer approaches that target ad hoc networks [100, 158, 160, 161, 166, 172] fall into 4 categories: TDMA ad hoc networks, UWB ad hoc networks, multimedia ad hoc networks, and general ad hoc networks.

TDMA Ad Hoc Networks

The main characteristic of a TDMA-based ad hoc network model is the notion of fixed time slots for channel separation. TDMA-based approaches in the literature also share the following features:

- **Synchronization:** all models assume that node clocks are synchronized. Synchronization is an essential component of any TDMA-based approach
- **Stationary nodes:** all of these models assume that nodes are stationary for their optimizations. Although this assumption grounds the optimization problem, it also limits the applicability of the proposed approaches to a wide range of mobile ad hoc network applications
- **Sender-initiated communication:** another shared assumption is source-oriented communication. This assumption models the intermittent communication trend in many ad hoc networks, as opposed to the regular data flow in monitoring or real-time networks
- **Transceiver power:** all models place a limit on the transmission power as a constraint for the optimization problem, focusing on determining the optimal transmission power for given link schedules

- **Omnidirectional antenna:** all models assume omnidirectional antennas, which is the simpler and cheaper choice. Directional antennas could enable further optimizations, but they add hardware complexity and cost

In addition to the common features of TDMA-based cross-layer approaches, individual models have made further assumptions for developing their optimization algorithm. For example, a functional requirement of most TDMA-based approaches is a dedicated control channel in the form of a reserved time slot [158]. The control channel enables reservation and scheduling of other time slots for data transmissions. This is a reasonable assumption since the only cost involved is the allocation of time slots for control messaging.

Another assumption that facilitates routing is the availability of geographic information at the nodes [160]. Nodes can narrow route searches to more specific geographic regions, thereby preserving resources (see Section 4.5). Providing nodes with geographic information requires additional positioning mechanisms at the nodes, such as GPS.

UWB Ad Hoc Networks

Two of the cross-layer approaches in the literature target UWB ad hoc networks [100, 172]. Both of these approaches target time-hopping UWB networks, and they perform their evaluations in a stationary environment. Additionally, both models consider the spread-spectrum nature of UWB transmission for determining the signal quality at a receiver, which is dependent on the active transmissions in the vicinity of the receiver. As such, both models tackle the problem of assigning the optimal transmission parameters of a new link while preserving the quality requirements of the active links.

What differentiates the two models are the assumptions of the tunable transmission parameter. Merz et al. consider that nodes always transmit with the maximum transmission power, and they regulate the transmission rate to achieve an acceptable SNR at the receiver. In contrast, Jurdak's approach considers two traffic types: reserved bandwidth (RB), and dynamic bandwidth (DB). The RB traffic class requires rate guarantees, so the approach adjusts transmission power while providing a constant transmission rate. The DB class supports best-effort traffic, enabling an elastic rate adjustment without changing transmission power. The DB traffic case of Jurdak's approach resembles Merz's model, while the RB traffic class provides support for higher priority traffic.

Another difference between the two approaches is that Jurdak assumes the presence of a separate TH-code for control signalling that is known to all nodes. Merz considers that each receiver owns its own TH-code, and that senders target the receiver by using the receiver's TH-code. While Merz's approach is more scalable with network size, it also requires nodes to learn and store the state of all potential neighbors. In dynamic or mobile networks, the overhead of this approach may overweigh the benefits.

Multimedia Ad Hoc Networks

One application for connection-oriented ad hoc networks is the transport of multimedia streams. The stringent rate and deadline requirements of multimedia applications complicate the task of running these applications over mobile ad hoc networks. Because ad hoc network nodes may enter, leave, or move around within the network, Chen et al. [167] propose cross-layer interaction between the routing and middleware layers to increase data availability for multimedia applications. The basic premise of their approach is that multimedia applications have higher priority over regular data, and different multimedia streams may have various degrees of priority among them. Providing the routing layer with visibility into the priority of multimedia streams enables delivery of urgent packets first. Similarly, the middleware layer can exploit routing layer information, such as node locations and mobility, in order to adapt data compression rates or to replicate data to ensure availability to multimedia applications.

General Ad Hoc Networks

Wang and Kar [166] consider the problem of maximization the bandwidth utilization for end-to-end sessions in multi-hop wireless ad hoc networks. This approach maintains generality as it makes no assumptions on time synchronization between the nodes or regarding the target application or underlying physical layer technology. The only assumption in Wang and Kar's approach about the target scenario is the symmetric hearing matrix, which assumes bidirectional links. In practice, assuming that all links are bidirectional limits the scope of the approach to ad hoc networks with highly reliable links and few physical obstacles.

9.4.2 Sensor Networks

Recently, researchers have proposed cross-layer approaches specifically for sensor networks [100, 162–164, 168, 171]. What distinguishes sensor network approaches from general ad hoc network approaches are the assumptions of a common network-wide goal, data flow towards one or few sinks, and mostly stationary nodes. Sensor network approaches fall into two broad categories: TDMA-based approaches and non-TDMA approaches.

TDMA Sensor Networks

Cross layer approaches for sensor networks that rely on a TDMA structure are suitable for networks with regular and frequent traffic. As in the case of TDMA-based ad hoc network approaches, TDMA-based sensor network approaches adopt a slotted time structure. Most of the other assumptions of TDMA-based ad hoc networks also apply for TDMA sensor networks. We note here that maintaining node synchronization in sensor networks could be even more costly in dense or dynamic networks, resulting in less efficient usage of the scarce node resources.

Non-TDMA Sensor Networks

Approaches for non-TDMA sensor networks [100, 168, 171] do not assume any time-slotted structure for transmissions. In general, these approaches are more suitable for long-term monitoring networks in which nodes sample their sensors and send data infrequently or asynchronously, switching into low-power sleep modes when they are not active.

Despite its support for asynchronous data transmission, Sichitiu's approach assumes that nodes synchronize their clocks for establishing sleep and wake-up schedules for transceivers. The authors also show that a decrease in synchronization precision does not significantly degrade their approach's performance.

Both the SP protocol and Jurdak's approach move away from the synchronization approach altogether. Both works adopt adaptive and asynchronous listening modes, ensuring that nodes can turn off their transceivers for most of the time while still hearing all relevant data and control packets. The key concept is to enable a sender to send packets with the appropriate preamble length to match the receiver's check interval. What distinguishes the two approaches are the factors that determine the preamble length. Whereas the SP protocol determines the preamble length purely on the channel state, Jurdak's approach combines aspects from across the network stack, such as a node's role and its number of descendants, in order to determine listening modes and preamble lengths.

9.5 Input Aspects

As in most algorithms, optimization decisions in cross-layer approaches rely on input information about the state of nodes, links, or the network as a whole. What distinguishes cross-layer approaches from strictly layered approaches is that the input state information comes from several layers across the communication stack. Consolidating state information across traditional layers enables deeper and more granular performance gains. Although cross-layer approaches advocate a shift away from strict layers, providing smooth transitions from traditional layered approaches to cross-layer approaches begins by mapping the information used in cross-layer approaches to the original layers in traditional approaches. Each subsection examines the input state variables of cross-layer approaches based on their layer classification in traditional layered architectures.

9.5.1 Application Layer

Cross-layer approaches that incorporate application layer variables into optimization decisions make inherent assumptions about the target application. From Tables 9.1 and 9.2, we note all the approaches that use input information from the application layer are information sharing approaches, and three of these four approaches target monitoring sensor networks [100, 171]. Monitoring sensor networks assume the presence of one or a few data sinks, and that sensors periodically sample and send data towards the data sink. The remaining approach that uses application layer information considers a technology-specific network (UWB) which entails implicit assumptions on the application, including a short-range high bandwidth scenario.

Current approaches also differ with their choice of application input state information. For example, Sichitiu's sensor network approach [171] allows generic applications to determine next hop information, without specifying the nature of the application variables. Jurdak's approach for UWB ad hoc networks [100] obtains the Mean Time Between Failures (MTBF) variable from the application layer. MTBF is a variable through which an application quantifies the communication and functional reliability of a node [175].

Other approaches use sensor-network specific variables. In particular, the work in [100] uses the node's data reporting frequency, available locally, as an input to the battery lifetime optimization method. Adaptive Low Power Listening [100] includes information about each node's *role* to determine the node's optimal check interval. For example, nodes that detect a fire in a forest-fire monitoring network adopt a *reporting* role and begin sampling their sensors more frequently. Such nodes can use role information for adjusting their listening modes.

9.5.2 Middleware Layer

One of the surveyed approaches [167] uses input information from the middleware layer for cross-layer optimizations. In their approach, Chen et al. provide the data priority information from the middleware layer to the routing layer to enable priority-based routing of data packets. The provision of middleware data priority information enables the network layer to establish routes that provide QoS guarantees.

9.5.3 Transport Layer

So far, there has been no documented use of transport layer information as input for cross-layer optimizations for ad hoc or sensor networks. However, one can envision scenarios that use transport layer information, such as congestion windows or timeout values to prioritize packet delivery at the routing or data link layer. As cross-layer approaches start to shift from algorithmic to architectural nature, we expect that there will be more cross-layer approaches that exploit transport layer information in the foreseeable future.

9.5.4 Network Layer

Half of the surveyed cross-layer approaches for ad hoc and sensor networks include network layer information as input into their optimization decisions. Network layer information can benefit end-to-end and per-link throughput, delay, and energy consumption.

In the simplest case, nodes in a tree topology network (which is common in sensor networks) can use information on their hop count from the tree root (the base station) for communication decisions. For example, nodes can use their basic hop count information along with specialized control messages for setting up non-conflicting link schedules in a sensor network [171].

A similar network layer metric that applies for tree topology networks is the *number of descendants* in the tree [100,163,171]. Because data flow in these networks flows towards the base station, a node's number of descendants in the routing tree indicates its forwarding load. Leaf nodes have no forwarding load, while nodes that are close to the base station have a relatively large forwarding load. Using descendant information enables nodes to adapt network behavior distributively to favor highly loaded nodes. For example, nodes can shift away their traffic from highly loaded nodes [100]. In an underwater network, nodes with higher load can use lower transmission frequencies, thereby reducing their power consumption [100].

Routing queue length is another network layer variable used in cross-layer optimizations. Girici and Ephremides [158] demonstrate that including the queue length into the routing metrics yields the best delay performance in their joint routing and scheduling optimization scheme.

In mobile networks, location and mobility information are important routing aspects that affect performance of several layers of the communication stack [167]. For example, mobility and location may affect collisions and channel access variables at the MAC, or data accessability at the application layer. Location and mobility are particularly important for multimedia applications that require a high degree of data availability to meet rate and delay guarantees.

Simple network layer attributes, such as reliability or urgency, can also play a role in determining the behavior of other layers. The network layer in SP [168] informs the link layer of message reliability or urgency merely by setting the corresponding bit values in available cross-layer data structures. SP's use of only two bits for communicating these network layer attributes can instruct the link layer on how to handle the messages without adding appreciable storage overhead.

Finally, nodes can exploit the end-to-end bandwidth requirements for optimizing their behavior. For example, Jurdak's approach for UWB ad hoc networks tags traffic as either reserved bandwidth or dynamic bandwidth traffic. Nodes can use the traffic rate guarantees (or the lack thereof) at the physical layer for adjusting the link transmission rate or the link transmission power.

9.5.5 Data Link Layer

The importance of the state of a nodes' wireless links has led researchers to include data link layer information into their cross-layer models. For example, the routing layer can exploit link layer information for choosing routes that provide throughput guarantees. Girici and Ephremides [158] incorporate a link's overall throughput during the network lifetime as a weighted routing metric. Their scheme favors outgoing links that can support high throughput routes. Each node can execute this cross-layer algorithm locally, yielding higher network throughput. A similar approach is to use a dynamic measure of link quality for determining optimal routes, as suggested in [171].

The average long-term *link* data rate is another data link layer metric for cross-layer optimizations. The approach in [159] uses each link's average data rate as an input to solve the joint scheduling and power control problem. In other words, this approach exploits cross-layer interaction between the average link data rate variable (a data link layer aspect) to determine, among other things, the transmission power configuration at the physical layer. Kozat et al. [161] provide a similar cross-layer coupling through their consideration of short-term *end-to-end* rates for choosing transmission power in attempt to provide a more adaptive model for ad hoc networks. Wang and Kar's approach [166] provides the option of slow or faster adaption of the session rates at the transport layer and the link rates at the data link layer to the existing link attempt probabilities.

In networks with adaptive duty cycles, the active and sleep schedules of nodes play a significant role in determining communication behavior. Two cross-layer approaches [100, 168] use the sleep and active schedule of neighbors in order to determine the preamble length for the BMAC protocol. The availability of neighbor schedule information locally enables a node to save on energy consumption, either through the use of short preamble packets or through the reduction of packet retransmissions caused by mismatches between the sender and receiver.

Finally, one cross-layer approach [100] has exploited the radio duty cycle information from the data link layer. This approach enables nodes to learn the radio duty cycles of its one-hop neighbors and to set neighbor routing costs in a manner that favors nodes with lower radio duty cycles. The new routing costs shift traffic away from busier nodes, enabling the busy nodes to put their transceivers in sleep mode more often.

9.5.6 Physical Layer

The most widely used input variables for cross-layer approaches originate at the physical layer. Cross-layer approaches can exploit physical layer information, such as transmission rate or transmission power, for guiding decisions at higher layers, such as link scheduling or routing choices.

Physical layer variables for cross-layer approaches fall into several categories: power consumption variables, transmission quality variables, and local node energy variables. Power consumption variables include power consumption for transmission, processing, sensing, as well as for other node hardware. Transmission quality variables indicate the physical layer quality requirements of a link, such as the desired SNR and the tolerated BER. Finally, the local energy variables include the residual battery energy at the node, and any potential energy harvesting gains.

Power consumption variables dominate the physical layer input variables. In particular, many approaches adjust the transmission power for cross-layer optimizations. One approach is to assume that each link uses the lowest transmission power possible to achieve error-free communications, as in [158]. Other approaches subject links to a maximum link transmission power constraint [100, 159, 165]. Interestingly, the approach in [172] remarks that optimality calls for using the maximum transmission power for every link, so it adopts a fixed transmission power strategy that always uses the highest transmission power.

Other examples of power consumption variables include processing, circuit and sensing power consumption. Circuit power consumption becomes a significant power sink for short-range networks in which transceiver power is relatively low [165]. As such, Cui's approach considers circuit processing power in its minimization of network power consumption. A similar reasoning for sensor networks [100] raises the need to consider the non-negligible sensing power consumption for energy optimizations.

Most approaches attempt to minimize power consumption without violating physical layer quality metrics, such as a minimum acceptable SNR or a maximum BER. ElBatt and Ephremides [160] develop their joint scheduling and power control algorithm to satisfy a set of SNR requirements. Another approach to the same optimization problem uses maximum BER constraints, assuming BER is monotonically decreasing with increasing SNR [161]. In addition to using SNR as a feasibility constraint for the solution of a cross-layer optimization problem, both algorithms in [162] and [163] also rank potential links according to their achievable SNR. As the algorithms progress, they attempt to generate a feasible transmission scenario from an infeasible scenario by deferring links with low SNR.

UWB networks have their own SNR definition which accounts for the interference from all concurrent transmissions. Because feasibility of new UWB links requires a minimum SNR for the new link and maintaining the feasibility of ongoing links, Jurdak's approach proposes that nodes establish links with a SNR that is higher than the minimum SNR by some margin. The margin is dependent on the network density, and it mitigates both bandwidth starvation and link quality violations.

Another approach for UWB ad hoc networks [172] considers the difficulty of computing the receiver SNR for a new link at the sender. Instead, Merz's approach enables a sender to progressively discover the best transmission time and optimal code rate through repeated transmissions on the receiver's TH-code.

Finally, Girici and Ephremides [158] consider the residual energy metric from the physical layer to guide routing decisions. Nodes with lower residual energy levels are avoided in routing paths. The drawback of using the residual energy metric, usually quantified by the battery voltage, is that small battery discharge trends involve fluctuations as the voltage level drops, which may lead to routing flaps.

This section has discussed the input variables considered by cross-layer approaches for ad hoc and sensor networks. Cross-layer approaches operate on the input variables to determine the optimal values for a set of variables and configurations. The next section examines these output variables and configurations.

9.6 Configuration Optimizations

Most of the existing cross-layer approaches have focused on optimizing configurations that span the first three layers of the communication stack, while research has been limited on adjusting configurations at higher layers for network optimization. The following factors explain this trend:

- The application layer typically sets the requirements for the rest of the layers. Thus, the application layer variables serve as inputs rather than outputs for the cross-layer optimization approach
- While there has been a body of research on transport protocols for ad hoc and sensor networks (see Section 5.1), the reliability, flow control, and congestion control mechanisms are not as crucial for the operation of ad hoc and sensor networks as lower layer functionality
- The network, data link, and physical layers constitute the core of network operation and interactions. The distributed management requirement of ad hoc and sensor networks has the deepest impact on these three layers, bringing them to the forefront of optimization efforts.

Each subsection in the remainder of this section discusses the optimization of network configurations at a single layer.

9.6.1 Middleware

Among the fifteen surveyed cross-layer approaches in this chapter, Chen's approach is the only one that optimizes the configuration at the middleware layer, violating the prevailing trend of lower layer optimizations. Through the use of location and mobility information from the routing layer, the middleware layer predicts network connectivity. As such, the middleware layer adapts data replication policies to ensure data is available for the active applications.

The approach also modifies middleware configuration based on current route quality. If the transmission quality over a certain route drops, then the middleware layer can adapt data compression rates to better fit the degraded route quality.

9.6.2 Transport Layer

Wang and Kar's approach [166] represents another example of cross-layer optimizations that target higher layers, through its joint optimization of the transport and data link layer configurations. In particular, their approach focuses on the optimization of the end-to-end session rates according to the observed neighborhood link rates. Nodes periodically measure their current link attempt probabilities and compute their link rates. These values are shared with the transport layer which can then proceed to compute the optimal end-to-end rates of active sessions on the basis of the current link conditions.

9.6.3 Network Layer

About half of the surveyed cross-layer approaches attempt to optimize the network layer configuration. All of these approaches set the routing costs in a manner that promotes their global performance goals.

The most common approach is the use of compound routing cost functions that combine several weighted route metrics [100, 158]. Weighted cost functions are tunable to fit different application needs, by adjusting the weights of the cost function components. In one approach, the cost function integrates energy-related metrics, such as radio duty cycle and sensing duty cycle, as well as hop count and link quality. The cost function can also take a more comprehensive definition that includes reliability, delay, and throughput from across the communication stack [100, 158].

Another class of approaches algorithmically derives the routing cost as a joint metric that embodies power and link scheduling considerations [159, 165, 171]. The resulting routing cost for each link is then used with a traditional routing algorithm, such as a shortest path algorithm, to determine optimal paths.

Yet another perspective is to prioritize packet forwarding at the routing layers according to higher layer priorities, as suggested by Chen et al. [167]. Because applications may differ in their quality-of-service metrics, such as end-to-end rate or delivery deadlines, the routing layer may use this application layer information to prioritize urgent packets over other packets in an attempt to meet the QoS requirements of all active multimedia streams.

Finally, nodes can adjust their route flow rates based on cross-layer information sharing, as suggested in SP [168]. Within the SP proposal, feedback notifications from the data link layer can inform the network layer of congestion or phase conflicts on particular links, enabling the network layer to adjust data rates or to shift the transmission phase for those links.

9.6.4 Data Link Layer

All but one of the surveyed cross-layer approaches optimize the configuration of the data link layer, highlighting the instrumental role of the data link layer for optimal behavior in ad hoc and sensor networks. Data link layer optimizations range from link schedules to the adjustment of duty cycles and timers.

Link schedules are an important output of cross-layer approaches that target TDMA-based networks. The main link schedule problem is to schedule as many concurrent transmissions as possible while satisfying transmission power constraints [159–162]. In some cases, routing considerations [158, 163, 171] or circuit power [165] also constrain the link schedule. Because TDMA approaches structure time into slots and frames, optimizing link schedules reduces to determining the optimal assignment of links into time slots within the problem constraints. One set of approaches attempts to progressively fill available time slots evenly, while another set tries to maximize the scheduled links in one time slot before trying to fill another slot.

Wang and Kar's approach [166] optimizes link attempt probabilities rather than link schedules, since it relies on a probabilistic asynchronous approach rather than time synchronization. This approach begins with the current link attempt probabilities as an input, and it iteratively computes the optimal end-to-end session rates and the corresponding new link rates. SP [168] is another approach that relies on asynchronous link scheduling. Through its message pool data structure, SP enables the link layer to forcefully schedule certain messages or to send packets out-of-order for meeting network layer quality requirements, such as reliability or urgency.

A prominent trend among cross-layer approaches that target UWB ad hoc networks is to dynamically optimize data link layer timer values according to network conditions. For example, the data link layer in Merz's approach [172] determines when and how to set back-off timers after probing the receiver's TH-code. In another example, nodes in Jurdak's approach [100] periodically recompute the value of the timer for the hello messages depending on the rate of change of neighbor states. In a completely asynchronous approach, ALPL [100] enables each node to determine its optimal low power listening mode according to its number of descendants in the routing tree, indicated by the forwarding load.

9.6.5 Physical Layer

Most cross-layer approaches also attempt to optimize the physical layer configuration. Adjusting transmission power, transmission rate, or modulation parameters are common techniques for physical layer optimization.

The majority of cross-layer approaches that target physical layer configuration enable nodes to tune their transmission power to satisfy constraints including long-term or short-term transmission rates [159, 161] or routing flows [158,171]. The basic premise is that each transmitter has a maximum allowable power emission, set by hardware limitations or regulatory bodies. The goal is the adjustment of transmission powers in the network, within the maximum transmission power constraint and the constraints from other layers, to achieve feasible power assignments for all transmissions.

Another direction is to consider the joint adjustment of transmission power and transmission rates to satisfy the higher layer constraints. For example, Jurdak's UWB approach [100] considers two traffic classes: a rate-guaranteed class, and a best-effort class. For the rate-guaranteed class, power can be adjusted to maintain acceptable transmission rates. In contrast, transmission rates are elastic for the best-effort class and they vary according to interference conditions to maintain a constant transmission power. Another example is the joint computation of transmission powers and rates given a set of fixed rate requirements [163]. Nodes can select the transmission rate and power for multiple outgoing links to deliver the data to the destination through multipath routes.

Alternative approaches manipulate the code rate at the transmitter to satisfy network constraints. Cui et al. [165] determine the upper and lower bounds of their Multiple Quadrature Amplitude Modulation (MQAM) rate, as well as the transmission power. The minimum modulation rate is constrained by SNR considerations, while the maximum transmission power sets the upper bound for modulation rate. In a similar approach that is designed for UWB technology, Merz [172] determines the maximum usable TH-code rate that mitigates interference at neighboring links. In another technology-dependent approach for underwater acoustics, Jurdak [100] adjusts the transmission frequency of acoustic transceivers to favor highly loaded nodes. This optimal transmission frequency choice depends on sensor update period, routing flows, as well as propagation loss considerations.

9.7 Implementation

One of the more practical aspects of cross-layer optimization approaches is the proposed implementation strategy. The implementation strategy specifies the internode and interlayer interactions, as well as computational details of supporting the proposed approach. Naturally, a deeper specification of implementation strategy results in a more readily applicable approach. This section classifies the existing cross-layer approaches into three categories based on their implementation strategy: unspecified, centralized, and distributed.

9.7.1 Unspecified

Some approaches focus on obtaining an optimal solution to a cross-layer problem without specification of any implementation details [14,158,162,163,165].

The main advantage of these approaches is that they typically focus on developing computationally efficient algorithms that can scale for large networks. The tradeoff for the focus on algorithmic development is that implementation details remain unresolved.

Most of the approaches with an unspecified implementation rely solely on theoretical analysis and design coupling without alluding to practical implementation aspects which are significant in the emerging research field of ad hoc and sensor networks. For example, these approaches derive polynomial or exponential order optimization algorithms that couple the functionalities of several layers. However, many nodes in ad hoc and sensor networks may lack the resources for running these algorithms, which leaves an open question about the applicability of these approaches in practical scenarios. Cross-layer approaches with unspecified implementation strategies also neglect the design of internode protocol interactions and the intranode layer interactions.

Most of these approaches make simplifying assumptions for their optimization algorithms. For example, several approaches [158, 162, 163, 165] assume node synchronization, but the absence of implementation details means that it is up to the network programmer to figure out the hardware and software requirements to achieve synchronization.

The performance evaluation of cross-layer approaches with an unspecified implementation strategy rely on numerical examples [165], numerical simulations [14, 162, 163], or queueing models [158]. Although this type of performance evaluation provides theoretical validation, the performance evaluation of these approaches does not address practical issues such as interference or dynamic deployment environments.

9.7.2 Centralized

Centralized implementation strategies specify that the optimization algorithm runs at a central controller [160,161]. Because of the lack of a central controller in ad hoc and sensor networks, developing cross-layer approaches for ad hoc and sensor networks with a centralized implementation strategy is certainly an interim step towards the development of fully distributed schemes.

Cross-layer approaches with a centralized implementation strategy assume that the central controller has a global view of the network state. By running the optimization algorithm, the central controller can notify each node of its optimal configuration. Of course, dynamic network conditions dictate that the collection of network state and the dissemination of node configurations is repeated periodically, which is not in a scalable solution for large networks. ElBatt and Ephremides [160] attempt to partially mitigate this issue by distributing power control computations to the nodes while running the scheduling algorithm centrally.

9.7.3 Distributed

Distributed implementation strategies are more suitable for ad hoc and sensor networks, as they spread the communication and computation among the nodes. Interestingly, five out of six applied information sharing approaches adopt a distributed implementation strategy, compared to only 33% of design coupling approaches.

Cruz and Santhanam's approach [159] provides limited implementation details for wireless base stations. This approach adopts a hierarchical clustered topology where link schedules of each cluster are determined independently, and then clusters collaborate to determine further dependencies. Although this approach has a distributed computational and collaboration strategy, it falls short of specifying the internode and interlayer interactions to support the distributed strategy. Taking a step further, Wang and Kar's approach [166] specifies the required interlayer interaction between the transport and the data link layers in order to optimize session and link rates respectively. This approach also supports distributed computation at each node, but it falls short of specifying the internode protocol interaction details in relation to the cross-layer optimizations.

Another class of approaches adopts a fully distributed strategy and specifies protocol and interlayer interactions. For example, Sichitiu's cross-layer approach [171] describes all the distributed computation, control messaging, phases for forming paths, scheduling links, and handling failures to maximize the lifetime of monitoring sensor networks. Chen et al. [167] adopt a fully distributed data advertising and lookup service, which, in conjunction with their middleware/routing cross-layer approach, supports data replication and ensures data accessability for multimedia applications. Similar fully distributed cross-layer approaches include SP [168] and ALPL [100], which both enable nodes to determine optimal listening modes locally based on local and neighbor states. The evaluation of both approaches was conducted on test-beds of sensor nodes, enabling a more realistic observation of these approaches in a practical scenario.

Other distributed cross-layer approaches targeting UWB ad hoc networks [100, 172] provide a similar specification of the control messaging, local node computation, and link failure and deferment to maximize network throughput and promote fairness. These approaches require nodes to learn partial state of the network in order to locally compute their optimal configurations. All of these distributed approaches demonstrate their applicability and performance through network simulations that model interlayer interactions and the internode interactions.

9.8 Conclusion

This chapter has surveyed existing cross-layer approaches for ad hoc and sensor networks. After a brief overview of each approach, the approaches were classified and analyzed according to five different features: (1) performance goals; (2) target network; (3) input aspects; (4) optimizations; and (5) implementation. The most common performance goals are energy-related aspects, such as network lifetime or energy efficiency. Cross-layer approaches have targeted different network types, including TDMA ad hoc or sensor networks, UWB ad hoc networks, underwater acoustic sensor networks, and RF sensor networks. Most of the input aspects originate from the first three layers, while some approaches also include application or middleware layer aspects as input to their optimization algorithm. The optimizations of almost all approaches are restricted to the physical, data link, and network layers. Finally, we have discussed the implementation strategy of the surveyed approaches, noting that distributed strategies are the most appropriate for ad hoc and sensor networks.

The next part of the book focuses on three applied diverse case studies of Jurdak's cross-layer optimization approach for different target networks: RF sensor network, UWB ad hoc network, and underwater acoustic sensor network. For each case, we specify the performance goals, input aspects, optimizations, and implementation strategy, before presenting the optimization algorithms, protocols, or heuristics.

Part III

Case Studies

10

Optimization of an RF Sensor Network

This case study applies Jurdak's framework to a test-bed of sensor nodes that communicate through RF signals. In particular, we explore the benefits of this approach in reducing the idle listening at individual nodes in order to reduce the energy consumption and to balance the forwarding load among network nodes. Our study introduces Adaptive Low Power Listening (ALPL), an adaptive cross-layer mechanism that builds on BMAC [56] in order to enable individual nodes to seamlessly adapt their listening modes to their local states. The chapter identifies three state aspects that impact energy consumption: (1) number of descendants in the routing tree; (2) radio duty cycle; and (3) role. We conduct experiments on a test-bed of 14 mica 2 sensor nodes to compare the state representations and to evaluate the framework's energy benefits. The experiments show that the degree of load balancing increases for expanded cross-layer state representations. The experiments also reveal that all state representations in Jurdak's framework reduce global energy consumption in the range of one-third for a time-driven monitoring network, and in the range of one-fifth for an event-driven target tracking network.

10.1 Introduction

Most conventional networks have many users, with each user having their individual application objectives. As a result, optimization in conventional networks has often focused on providing performance guarantees on a per-user or per-connection basis. In sensor networks, the nodes collaborate to achieve a common network-wide goal. As such, sensor networks must provide global network performance guarantees instead of providing user-specific or node-specific performance guarantees, suggesting the need for global optimization mechanisms.

Portions reprinted from (R. Jurdak. "Modeling and Optimization of Ad Hoc and Sensor Networks," Bren School of Information and Computer Science, University of California Irvine. Ph.D. Dissertation. September, 2005)

In this chapter, we explore the benefits of optimizing global network energy consumption through greedy local decisions made by each node based on its local state. In particular, the nodes adapt their routing and MAC layer behavior to their local state. Figure 10.1 shows Jurdak's customized framework for this case study. Enabling individual nodes to adjust their behavior according to their local state requires underlying mechanisms that are adaptive, flexible and modular. Figure 10.2 shows the communication mechanisms that drive the optimizations in this chapter.

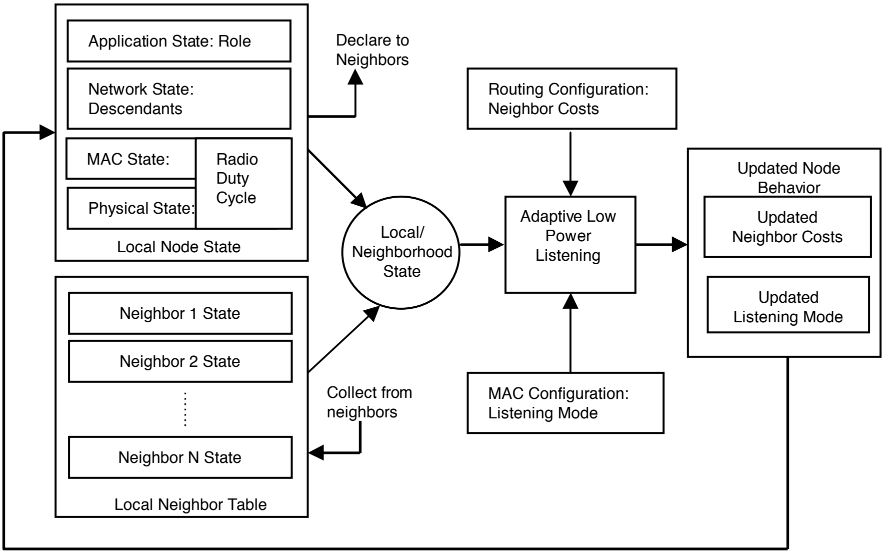


Fig. 10.1. RF cross-layer optimization framework, with ALPL serving as the local optimization algorithm

At the routing layer, nodes can learn the state information of their neighbors and set each neighbors' routing cost accordingly. The flexible and cross-layer state representation ensures that routing behavior in the network adapts to all relevant state parameters. Our model proposes a flexible cost function for global optimizations, inspired by the work of Baldi et al. [175]. The cost function supports all possible routing cost metrics of interest and is customizable to the performance requirements of each application. The cost function metrics include power terms related to the duty cycle of node radios and sensor activity. Additional terms in the cost function can be tailored to take into account QoS, for instance, by controlling the total number of hops along a communication path and, hence, the corresponding delay. Minimization of the cost function determines the routing strategies of the network.

At the MAC layer, nodes can also adapt their behavior according to the cross-layer state representation. Radio energy consumption is one of the main

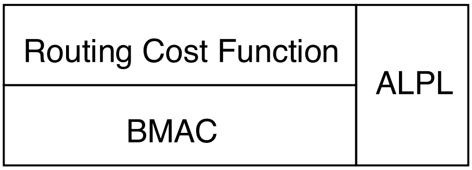


Fig. 10.2. Communication mechanisms

energy consumption sources at sensor nodes. In monitoring networks, idle listening is the main contributor to radio energy consumption. In order to minimize idle listening, each node can adapt its radio duty cycle according to its current local and neighborhood states.

We choose BMAC [56], a modular and flexible sensor network MAC protocol which aims at reducing idle listening at sensor nodes, to validate Jurdak's framework. BMAC enables each node to wake up periodically to check for channel activity. The wake-up period is referred to as the check interval. BMAC defines 8 check intervals, and each check interval corresponds to one of BMAC's 8 listening modes. To ensure that all packets are heard by the nodes, packets are sent with a preamble whose reception time is longer than the check interval. BMAC therefore defines 8 different preamble lengths referred to as transmit modes.

BMAC also provides interfaces that enable services and applications to set low power listening modes and transmit modes on a per-packet basis if needed. The creators of BMAC also suggest that further energy savings could be produced by using these interfaces to set listening and transmit modes according to additional information on the application and operation of a sensor network.

Building on BMAC, our work proposes a cross-layer mechanism called Adaptive Low Power Listening (ALPL) to provide a seamless basis for locally setting the listening mode while ensuring reliable data delivery. In original BMAC, setting a network-wide listening mode disregards the non uniform and dynamic local states of individual nodes. Per-node listening modes are more energy-efficient, but it is difficult to predict the state of each node prior to deployment. To address these challenges, ALPL supports the adaptation of listening modes in BMAC to local sensor node states, and it enables a node to learn the listening mode of its neighbors in order to ensure correct data delivery.

The rest of the chapter is organized as follows. Section 10.2 discusses related work, including BMAC. Section 10.3 describes ALPL in detail. Section 10.4 provides the analytical justification for reducing global cost through local decisions in ALPL. Section 10.5 presents our experiments to validate the approach on mica 2 motes. Section 10.6 discusses the deployment results.

10.2 Related Work

This section surveys the related work in the literature. The discussion first examines existing approaches on cost optimization. The second part of this section revisits approaches for providing energy efficiency at both the network and data link layers.

10.2.1 Cost Optimization

Previous efforts [175, 176] have realized the need for optimizing global cost in wireless networks, which determines the routing strategies of the network. Baldi et al. [175] develop a global cost function for wireless ultra wide band (UWB) radio networks, based on the cost of individual links. Their cost function is additive and considers transmission power, link setup cost, interference, delay and reliability. Mhatre et al. [176] optimize the hardware cost of heterogeneous sensor networks with a lifetime constraint. Their work considers a network with 2 types of nodes, and attempts to determine both the number and position of each node to minimize overall hardware cost while satisfying the lifetime constraint. Jurdak's customized framework focuses on dynamic cost optimization, so it disregards the hardware cost which is static once the network is deployed. The framework adopts a cost function similar to the function in [175] and customizes it for sensor networks through the introduction of the composite metrics of data delivery and energy efficiency that are central to most sensor network applications.

10.2.2 Energy Efficiency

In sensor networks, energy efficiency and load balancing are the primary metrics of interest. Many protocols have been designed to provide energy-efficient behavior at both the MAC layer [179, 180] and the routing layer [181].

Routing Protocols

As indicated in Chapter 4, routing protocols are either proactive [182, 183] or reactive [184, 185]. Proactive protocols maintain network or neighborhood routing state tables at each node. Reactive protocols compute optimal routes on-demand. Proactive protocols are advantageous for networks that require limited mobility, low latency, or high throughput. In contrast, reactive protocols are suitable for high mobility, low throughput, high latency networks. In this chapter, we consider a typical stationary time-driven monitoring sensor network in which the periodic nature of data transmission fits well with periodic state exchanges in proactive routing protocols. As a result, we consider a proactive and periodic routing protocol in which nodes store the routing state of their one-hop neighbors.

Although many cost metrics have been proposed for routing in sensor networks, we focus the discussion here on metrics for optimizing energy consumption. Previous work has examined energy optimizing routing strategies, where cost metrics include residual battery energy [186,187]. The residual battery capacity strategy is intuitively valid, but the discharge of real batteries becomes nonlinear or unpredictable at a certain voltage level, which effectively cancels battery considerations from routing decisions at some point during the deployment [101]. In our work, we consider the radio, sensor, and processor duty cycles in the recent time window as the energy metric for routing. We believe that this metric is more expressive of each sensor node's energy profile since it is independent of the battery technology and it does not rely on unpredictable battery discharge models.

MAC Protocols

At the MAC layer, idle listening constitutes a large portion of energy consumption because data is sent infrequently. This effect is even more pronounced in monitoring sensor networks [188]. Thus, energy-efficient MAC protocol proposals have focused on minimizing idle listening at sensor nodes [56–58, 189].

IEEE 802.15.4 [189] is a standard with physical and MAC layer specifications for low rate, low power, short-range networks, including sensor networks. IEEE 802.15.4 specifies a plethora of functionality choices, many of which may never be used. As a result, several researchers have proposed new MAC protocols on top of the 802.15.4 PHY layer.

SMAC [57] is a heavyweight MAC protocol for sensor networks that relies on time synchronization and scheduling among nodes to enforce periodic sleep and listen schedules. SMAC reduces energy consumption and provides scalability at the cost of per-hop fairness, throughput, and latency. A more recent version of SMAC [190] has introduced adaptive duty cycles by enabling a node to snoop on neighbors' RTS and CTS messages in order to schedule its own wake up time. Because nodes have to maintain neighbors' schedules, SMAC is complex and not sufficiently scalable for large scale networks of resource-limited nodes.

T-MAC [58] also proposes adaptive duty cycles to address the nonuniform traffic patterns in sensor networks. T-MAC is similar to SMAC in its essence, but it introduces early sleeping to enable nodes that are scheduled to be active to go into sleep mode if they are idle. T-MAC suffers from similar complexity and scaling problems of S-MAC, because it trades off a short active time for reduced adaptivity to changing network conditions.

The recent work by Polastre et al. proposes BMAC [56], a lightweight sensor network MAC protocol that aims at providing versatile medium access while keeping the MAC functionality as simple as possible. Because it is an asynchronous protocol, BMAC eliminates the communication and processing overhead for scheduling and synchronization, which reduces energy consumption. BMAC enables each node to wake up periodically to check for channel activity, where the wake-up period is referred to as the check interval. Polastre et al. analytically derive optimal check intervals based on the number of neighbors of a node. In their experiments, they determine the maximum neighborhood size in the network, and they set the optimal check interval (or listening mode) for that neighborhood size. The experimental results yield significant energy savings for BMAC over previous protocols, such as SMAC. BMAC is also the standard MAC protocol in the communication stack of TinyOS [191], the primary sensor network research platform.

As in T-MAC and SMAC, ALPL addresses the nonuniform node states by adapting listening modes in BMAC to enable adaptive duty cycles. We choose BMAC because of its flexibility and scalability, which are two of our main design goals. Instead of proposing a new MAC protocol, this chapter's focus is designing a mechanism (namely ALPL) that enables each node to adapt its duty cycle based on flexible cross-layer node state representations.

10.3 Adaptive Low Power Listening

The general algorithm within Jurdak's framework periodically runs the same three basic steps for any network scenario:

1. Gather neighborhood state information

2. Perform local calculations on gathered state

3. Modify local configuration accordingly

The details of the algorithm and its implementation are highly dependent on the network scenario. As such, this section discusses the details of the algorithm tailored for a monitoring sensor network application. We begin by introducing the motivation for ALPL within the framework. Next, we describe the node interaction that enables nodes to set listening modes through ALPL and to exchange their state information. We subsequently describe the three state representations that are considered in this chapter. Next, we explain how each node uses its neighborhood state information to calculate the routing cost of each neighbor through the cost function. Finally, we present the necessary routing protocol modifications to ensure correct data delivery with ALPL.

10.3.1 Adaptive Low Power Listening

Adaptive Low Power Listening [192, 193] is a cross-layer mechanism that adapts the listening mode at each node according to its local state, while ensuring correct data delivery.

The motivation for ALPL is to address the unpredictable and dynamic node states in sensor networks [195] which are affected by factors such as interference variations and dynamic node membership. Network designers have to make conservative assumptions in determining network configuration. In the case of energy optimization, conservative assumptions lead to setting a network-wide listening mode in BMAC prior to network deployment. This causes unnecessary idle listening to occur in less active portions of the network. ALPL's purpose is to reduce idle listening in BMAC by allowing each node to set its own listening mode depending on its local node and neighborhood states. The rationale is that in dynamic sensor networks, each node always has the most up-to-date view of its own local state [196]. Node states can be defined by the network designer or operator, depending on the applications' goals and quality-of-service requirements.

10.3.2 Node Collaboration

The internode collaboration strategy in Jurdak's framework enables a node to learn its neighborhood state information, to set its own listening mode accordingly, and to adapt its transmit mode to fit the listening mode of its routing parent. Figure 10.3 illustrates the node interaction to enable nodes to set their listening mode adaptively. We assume a proactive routing protocol in which nodes periodically send routing update messages to declare their routing information to their neighbors.

Initially, nodes are unaware of their neighborhood state, so all nodes listen at an initial listening mode L_{init} and use the corresponding transmit mode T_{init}, both of which are known a priori to all nodes. Each node begins sending periodic route update messages to declare its presence and state. Once nodes learn of their neighbors' presence, a routing graph is formed and data flows towards the base station (Figure 10.3(a)). As a result, each node learns the state of its direct neighbors. Before sending the next route update message, node A first sets the optimal listening mode L_A for its local state [1]. Then, node A sends a routing update message that includes its new listening mode and state information along with other routing information (Figure 10.3(b)).

All of A's neighbors hear the routing update message, and they learn A's current listening mode L_A and A's state information. Each neighbor of A records A's listening mode and state information in its local neighbor table. Consequently, each node in the network always has up-to-date information on the state of its neighbors. Whenever node D chooses A as a routing parent, it simply checks its neighbor table for A's listening mode L_A. D then sends its data packets using the transmit mode T_A that matches L_A. Similarly, nodes that receive a routing update message from their current parent indicating that the parent has a new listening mode adapt their transmit mode accordingly.

---

[1] Section 10.4.2 provides a detailed example of how a node can locally select the optimal listening mode

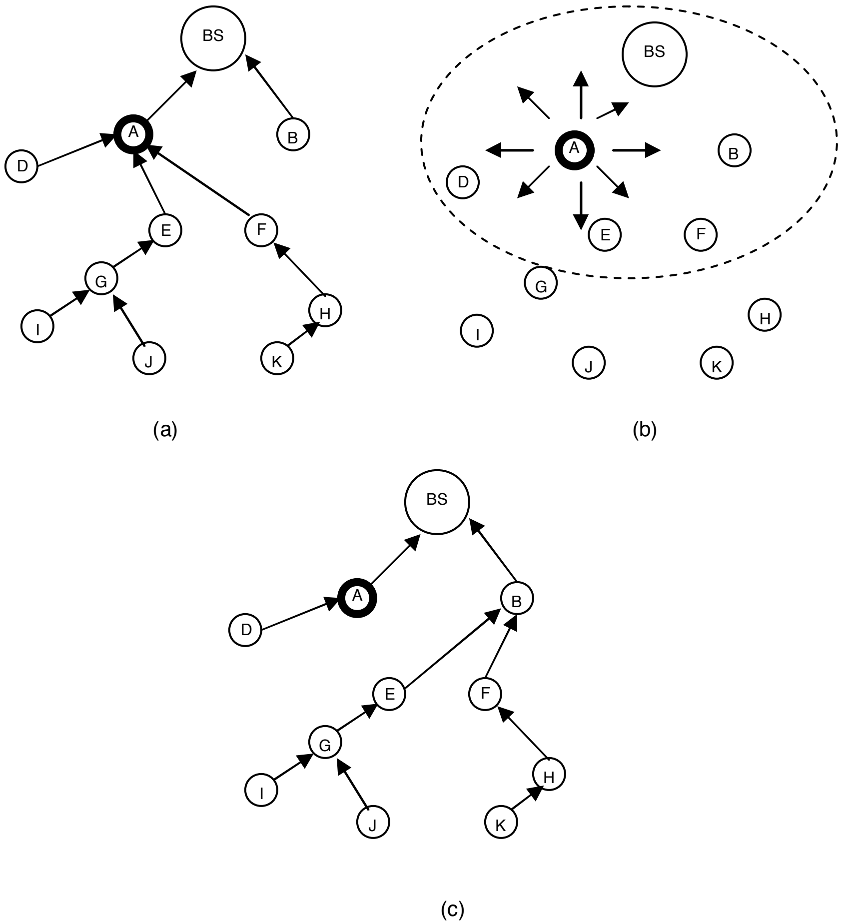


Fig. 10.3. (a) Nodes form their routing tree. (b) Each node periodically announces its current state, including its listening mode and battery state, enabling neighbors to use the appropriate transmit mode. (c) A high duty cycle at A causes neighbors to increase A's routing cost and choose a new parent B. Node A listens less often as it has fewer packets to forward

For a particular node in ALPL, the only concern for data delivery is whether the routing parent is reachable. If so, then the node can transmit with the proper preamble length. Otherwise, the node detects the unreachability of the routing parent through missed route update packets and attempts to find another suitable routing parent. If the node hears a routing update packet from the original routing parent at some point, the node can rerun the routing decisions for choosing the routing parent and the appropriate listening mode.

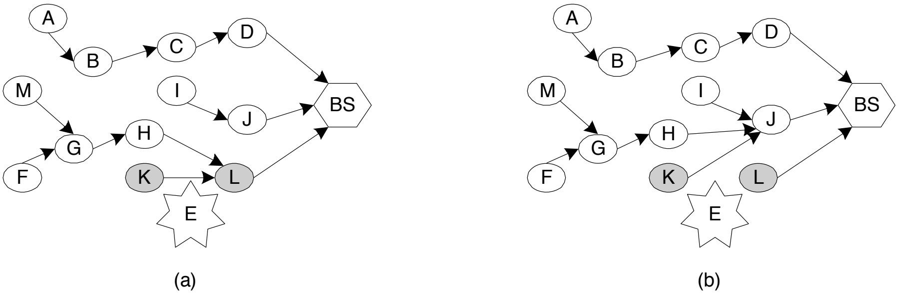


Fig. 10.4. An event-driven tracking sensor network

10.3.3 State Representations

Our state representations consist of three aspects: (1) number of descendants; (2) duty cycle; and (3) role. In this subsection, we describe each of these 3 aspects in detail.

Number of Descendants

Our first representation of state considers a node's number of descendants in the routing tree, which has significant impact on a node's energy profile in data gathering and monitoring applications. In monitoring applications, data flow is typically towards a single data sink. The basic requirement for correct data delivery is for each node to listen often enough to hear all the packets that it must forward toward the data sink. The number of packets that a node forwards depends on the number of its descendants in the routing tree.

In order to determine its optimal listening mode, each node N learns how many descendants it has in the routing tree by counting the number of packets γ that it forwards during a route update interval. The number γ indicates how busy N was during the last interval. When it is time to send the next routing update message, N first sets its listening mode to the optimal listening mode L_N for a traffic load of γ packets. Then, N sends a routing update message that contains L_N along with other routing information.

Including the number of descendants into ALPL's state representation optimizes the overall energy consumption in the network. Each node optimizes its local energy consumption through the selection of an optimal listening mode for its current forwarding load. The distributed optimization of local node energy consumption leads to an optimization of network energy consumption. The number of descendants only exploits the load imbalance in the network to opportunistically optimize energy consumption at lightly loaded nodes. The next two sections present additional node state aspects that promote load balancing.

Duty Cycle

Our second state representation combines the number of descendants and the node's duty cycle. A high radio duty cycle indicates that the node's radio has been highly active up until the present point in time and vice versa. Given the inherently nonuniform energy consumption in sensor networks, enabling nodes to adapt their behavior according to their radio duty cycle can help balance the energy consumption in the network.

Sharing duty cycle information among nodes involves minimal overhead communication. Each node can piggyback its duty cycle value within its routing update messages in order to declare its power state to neighbors. As a result, nodes learn the radio duty cycles of all their one-hop neighbors and store this information in their local neighbor table.

Including the radio duty cycle into ALPL's state representation enables local load balancing of network traffic. Each node can consider and compare the recent activity level of neighbor transceivers. Neighbors with a relatively higher radio duty cycle are penalized through higher routing cost, which diverts traffic away from busier nodes to other nodes to achieve more uniform energy consumption. The radio duty cycle promotes load balancing to the extent allowed by the routing graph. As long as the routing graph provides a node with more than one alternative for a routing parent, the radio duty cycle information contributes favorably to load balancing.

As an example, consider Figure 10.3 again. The routing tree in Figure 10.3(a) puts most of the forwarding burden on node A. As a result, A depletes its battery resources quicker than node B. In order to shift its forwarding load, A declares its high duty cycle to its neighbors, causing neighbors to increase A's routing cost. This in turn causes most of A's current children to choose another parent whenever possible (Figure 10.3(c)). Having diverted most of its forwarding load to node B, node A begins listening with a longer check interval to reduce its listening energy consumption.

Role

The nonuniform state of sensor nodes also stems from the roles that the nodes may take during deployment. The number of descendants is a routing aspect and it represents the present state of the node. The duty cycle is a physical and MAC layer aspect that represents the past state of the node. The node's role provides information on application functionality and represents its projected state into the future. In the context of our energy optimization study, the node's role can play a part in determining the node's optimal listening mode and neighbors' routing costs. The rationale is to incorporate additional knowledge about the expected power profile of a node's role in optimal local decisions.

In order to integrate role into network decisions, nodes should share their role status with their neighbors. Sharing role information among nodes involves minimal overhead communication. For example, in a network where nodes may take one of two roles, each node can include its role status within its routing update messages through a single bit. As a result, nodes learn the roles of all their one-hop neighbors and store this information in their local neighbor table.

As in the case of radio duty cycles, including node roles into ALPL's state representation promotes local load balancing of network traffic. Each node can consider and compare the recent sensing activity level of neighboring nodes. Thus, a node can divert traffic away from nodes with high sensing activity, which contributes to load balancing. In homogeneous networks where all nodes behave the same, the role variable does not contribute to load balancing. However, in event or query driven networks, certain nodes may assume more active sensing roles during the deployment. In these cases, the energy consumption for sensing becomes more significant and it should be considered for load balancing considerations.

Consider the network in Figure 10.4. While nodes are periodically sending their data with period T, much of the forwarding load is focused at node L. At some point during the deployment, an event E occurs at the lower periphery of the network. Nodes K and L detect event E, switch their role to tracker, and begin sensing and reporting data at $T/2$. The neighbors of K and L learn that these 2 nodes are now tracker nodes and thereby increase the routing cost of these 2 nodes. Figure 10.4(b) shows the resulting topology. Nodes K and H choose a new parent, node J, in order to avoid using the tracker node L as a forwarder. The topology in Figure 10.4(b) puts most of the forwarding load on node J. The energy consumption resulting from the high forwarding load of node J is balanced by the energy consumption for more frequent sensing and data reporting at node L.

10.3.4 Cost Function

Our primary platform for sensor network development is TinyOS [191], developed at UC Berkeley. Within TinyOS, the standard routing protocol is called MintRoute [197]. MintRoute is a proactive routing protocol in which nodes send periodic routing messages to declare their local states to their one-hop neighbors.

In all proactive routing protocols and particularly in MintRoute, each node periodically selects its routing parent. Node N first selects the highest contender M in its neighbor table with the least routing cost at the current time. Next, N compares the cost of M with the cost of the current routing parent RP. N chooses M as its new parent only if:

$$C_{mint}(M) + \epsilon < C_{mint}(RP) \tag{10.1}$$

where $C_{mint}(N_j)$ is the routing cost of node N_j, and ϵ is the switching threshold that ensures that a node switches its routing parent only when there is an appreciable benefit in doing so.

The original cost metric in MintRoute involves both hop count and link quality. In particular, the cost metric assigns the link weight as the product of the reciprocal of the forward and backward link qualities, similar to the ETX routing metric in Section 4.1.2. MintRoute then uses a shortest path algorithm for determining the best route to a destination.

The main purpose of integrating a shortest path algorithm with a link quality metric is to maximize data delivery by choosing the shortest and most reliable path to the base station. The shortest path algorithm reduces the number of reliable nodes involved in forwarding activity. Link quality reduces the number of retransmissions by favoring more reliable links. The resulting algorithm also implicitly enforces energy efficiency. Since Jurdak's customized framework aims at making routing decisions dependent on the node states, we introduce additional cost metrics that are direct functions of the aspects that constitute node state. A discussion of further potential cost function metrics is available in Appendix A.

Duty Cycle Awareness

The number of descendants aspect is considered for setting local listening modes rather than setting neighbor costs, so this aspect does not require an explicit routing cost metric. For the duty cycle aspect, we introduce the cost metric $C(radio)$, which denotes how busy a particular neighbor's radio is relative to other neighboring nodes. Highly loaded nodes should have a higher routing cost. Therefore, we express $C(radio)$ for a neighbor N_j as:

$$C(radio) = \epsilon \frac{\delta_j - \sum_{i=0}^{k} \delta_i / k}{\sqrt{\sum_{i=0}^{k} \delta_i^2 / k - (\sum_{i=0}^{k} \delta_i / k)^2}} \tag{10.2}$$

where k is the number of neighbors in the node's local neighbor table, and δ_i is the radio duty cycle of neighbor N_i. Equation 10.2 compares the radio duty cycle of neighbor N_j to the average duty cycle in the neighborhood and normalizes the difference. The normalized difference determines the extent of statistical deviation of the radio activity of N_j among all the nodes in the neighborhood.

Role Awareness

The duty cycle is a sufficient indicator of energy profiles in cases where the nodes all have the same sensing and processing patterns throughout the lifetime of the network. In many event-driven or demand-driven sensor networks, some nodes change their application behavior, such as the sensing frequency or buffering strategy, based on events in the network. Such cases require the inclusion of information on the current roles of nodes in routing decisions.

The actual cost dependance on the node's role is highly application-specific. In this chapter, we consider an event-driven network that specifies that nodes can have one of two sensing frequencies f_1 and f_2, where f_2 is double f_1. A node uses the following equation to evaluate the sensing cost of a neighbor N_j:

$$C(sensing) = \begin{cases} \epsilon \frac{\theta_j - \sum_{i=0}^{k} \theta_i / k}{\sqrt{\sum_{i=0}^{k} \theta_i^2 / k - (\sum_{i=0}^{k} \theta_i / k)^2}} & \theta_j = 1 \\ 0 & \theta_j = 0 \end{cases} \tag{10.3}$$

where k is the number of neighbors in the node's local table. θ_i is a Boolean variable that takes the value of 1 if N_i is sensing data at frequency f_2 (tracker node) and takes the value of 0 if N_i is sensing data at frequency f_1. Equation 10.3 compares the sensing activity of neighbor N_j to the average sensing activity in the neighborhood and normalizes the difference. The normalized difference determines the extent of statistical deviation of the sensing activity of N_j among all the nodes in the neighborhood.

Overall Cost Metric

The overall routing cost should strike a balance between the need for correct and timely data delivery on one hand and uniform energy cost on the other. Furthermore, $C(radio)$ and $C(sensing)$ should play a role in determining the routing parent only if the node M is at the same or at a lower level than the current routing parent RP.

Therefore, the new overall cost of a neighbor includes the original MintRoute cost of quality and hops, as well as the power cost according to the following equation:

$$C(M) = \begin{cases} C_{mint}(M) + \alpha C(radio) + \beta C(sensing) & H(M) \leq H(RP) \\ C_{mint}(M) & H(M) > H(RP) \end{cases} \tag{10.4}$$

where α and β are constants representing the weights of $C(radio)$ and $C(sensing)$ respectively. $H(M)$ indicates the hop count of node M from the base station. With the new cost definition, neighbors with higher energy consumption (due either to radio duty cycle or to increased sensing activity) have a higher routing cost, and they are less likely to be chosen as forwarders.

Global Network Cost

The global network cost in Jurdak's customized framework is the sum of the costs of individual nodes [175]:

$$NC = \sum_{N} C(N) \tag{10.5}$$

Since optimizing NC is not feasible, each node selects its routing parent as the node with the least routing cost $C(N)$.

10.3.5 Routing Modifications

Enabling each node to set its own listening mode adapts MAC protocol behavior to the node's state. The MAC protocol adaptation should be accompanied with routing protocol adaptation in order to maximize performance gains.

The concept of adaptive listening modes raises the possibility that some nodes may not hear the packets sent by their neighbors because of mismatched preamble lengths and check intervals. For example, if node A sends a packet with a short preamble to its parent B, one of node A's neighbors C that is listening infrequently may miss node A's packet. This situation does not affect data delivery, since it is only necessary for A's parent B to hear the packet. Missing a routing update packet is more detrimental, since routing packets hold important information on neighborhood routing state changes.

We implement modifications to MintRoute to address missed routing update packets. A central issue in designing ALPL is to ensure that symmetric listening modes do not affect maintaining an up-to-date neighborhood view at each node. Achieving this goal requires that nodes always hear the routing update packets of their neighbors. Thus, ALPL specifies that nodes always send their routing update packets with the longest preamble, so that a neighbor in any listening mode can hear the update packets. Secondly, MintRoute determines quality as the percentage of data packets correctly received from a neighbor. The example above on missed data packets causes the quality metric in MintRoute to drop. Consequently, we modify MintRoute so that nodes only snoop on periodic routing updates instead of data packets to determine the link quality to their neighbors. Monitoring route update packets for determining link quality ensures that asymmetric listening modes at neighboring nodes have no detrimental effect on link quality, because all routing update packets are sent with the longest preamble.

10.4 Qualitative Analysis

This section presents the analytical basis for using greedy local decisions to reduce global network energy consumption. We assume that the sensor nodes collect sensor data and transmit the data in a packet once in every period T. The following equation governs the energy consumption E at a sensor node [56]:

$$E = E_t + E_r + E_d + E_{listen} + E_{sleep} \tag{10.6}$$

where E_t is the energy consumed on transmissions during time T, E_r is the energy consumed for packet reception during time T, E_d is the energy consumption to collect sensor values, E_{listen} is the energy consumed for checking the channel for activity, and E_{sleep} is the energy consumed while the node is asleep. The quantitative expressions for each energy component are given in [56]. We limit the discussion here to the qualitative aspects that are relevant to ALPL.

In monitoring and data gathering applications, the sampling period T is typically in the order of minutes. Therefore, each node collects sensor data, transmits packets, and receives packets once every few minutes. On the other hand, nodes wake up to monitor the channel for activity much more frequently, for instance once every several milliseconds. Thus, idle listening on the channel has a profound effect on the average power consumption, so reducing idle listening yields significant energy savings.

10.4.1 Topology

The aim of the number of descendants metric is to minimize the energy consumption in the network through local optimization decisions. Each node can optimize its own consumption E locally by selecting its own optimal listening mode while maintaining correct and timely data delivery. Minimizing E on a per-node basis reduces the overall network energy consumption and builds on the following observations about Equation 10.6:

1. E_d and E_{sleep} are not significant factors in determining optimal listening mode in a homogeneous monitoring network. E_d is equal for all nodes. E_{sleep} is at least an order of magnitude smaller than the other terms in Equation 10.6, so it has a negligible effect on E.
2. E_t and E_r depend on the node's position in the logical topology. If a node is a leaf in the routing tree, it has fewer packets to forward.
3. The listening mode N determines E_{listen}. It also determines the preamble length for packets that are received at N. Consequently, the listening mode at node N affects E_r at N and E_t at N's children.

For example, a more frequent listening mode at N increases E_{listen} and decreases E_r at N. E_{listen} increases because N wakes up more frequently to check for channel activity, and E_r decreases because the frequent listening enables N's neighbors to send their packets to N with shorter preambles, thereby reducing packet reception time at N.

Similarly, the listening mode of N affects E_t at the children c_i of N. A more frequent listening mode at N enables c_i to send its packets with shorter preambles, thus reducing E_{t_i}. Through similar reasoning, less frequent listening at N forces c_i to send packets with long preamble and consume more energy for packet transmissions.

These dependencies further support the need for setting per-node listening modes. In practice, each node locally computes E_t, E_d, and E_{sleep} and then selects the listening mode that provides the combination of E_{listen} and E_r that yields the lowest energy consumption E.

ALPL builds on the need to reduce idle listening and to balance energy consumption. ALPL's most basic state representation only considers number of descendants in the routing tree, which yields energy savings at nodes with few descendants. We now consider a case study analysis to illustrate the benefits, scalability and shortcomings of this most basic form of ALPL.

10.4.2 Case Study

To analyze the tradeoffs involved in ALPL and to compare the energy consumption of ALPL and BMAC, we consider a case study of a static 127-node network with a binary tree topology. Although the topology of an actual sensor network can be both irregular and transient according to environmental conditions as well as location, this case study serves the purpose of validating the analytical basis and the scalability of ALPL. In the next section, we perform experiments with actual sensor nodes to further validate our model in a more dynamic and realistic scenario. Our analysis for this network assumes that the sensor nodes sense the environment and send their data to the base station once every 2 minutes.

In their evaluation of BMAC, Polastre et al. use the following method for assigning a listening mode that favors the busiest node to improve network lifetime:

- Compute the expected number of descendants of the busiest node in the network
- Set the network-wide listening mode to favor the busiest node

Since the busiest node in the network has the largest forwarding load, this method for setting listening mode typically means that all the nodes in the network use the higher duty cycle setting which suits the busy node. This gives rise to one of the motivations of ALPL: to enable nodes that are not as busy to choose their appropriate duty cycle in a decentralized manner.

Table 10.1 compares the check interval between network-wide listening modes (plain BMAC) and ALPL at each level in the 127-node binary tree network. In plain BMAC, the check interval is initially set to 10 ms for all the nodes in order to accommodate the forwarding load of the busiest node. In ALPL, the busiest node (the node at level 1) selects the same listening mode as BMAC nodes, and the remaining nodes select their own listening modes based on their topology position. The following example illustrates the listening mode selection process in ALPL.

We consider how a level 2 node in the 127-node binary tree selects its optimal check interval in ALPL. A level 2 node sends 63 data packets each update period, where 1 packet is locally generated and the rest are forwarded packets from its descendants. The level 2 node also uses a preamble length of 28 bytes in order to match its parent's check interval of 10 ms. The resulting ALPL energy consumption E_t during an update period for a level 2 node is 0.1006 Joules. Using the fact that it receives 62 data packets from its descendants each update period, the level 2 node then computes its reception energy consumption E_r for each of the 8 possible check intervals. It also computes E_{listen} and E_{sleep} for each check interval. Finally, the level 2 node selects a check interval of 20 ms, with a reception energy consumption E_r of 0.1509 Joules and a listening energy consumption E_{listen} of 0.1038 Joules, as the optimal check interval that yields the minimum overall energy consumption E of 0.3553 Joules during an update period at the level 2 node.

All other nodes in ALPL use the same process to select the optimal listening mode. As mentioned earlier, the level 1 node selects a check interval of 10 ms as in the BMAC case because it is the busiest node in the network. Nodes at level 2 and at higher levels choose less frequent listening modes because they have a smaller forwarding load than the level 1 node.

We note here that as the network size grows, more nodes at lower levels of the tree converge to the generic BMAC behavior. For example, if we double the network size of this case study to 256 nodes, nodes at level 2 choose their optimal check interval as 10 ms. Doubling the network size further causes level 3 nodes to select the highest duty cycle, and so on.

Level	1	2	3	4	5	6	7
BMAC Check Interval (ms)	10	10	10	10	10	10	10
ALPL Check Interval (ms)	10	20	20	50	100	200	1600

Table 10.1. A comparison of the listening modes at each level in the network

Figure 10.5 compares the sources of energy consumption for the two cases. E_d is omitted from Figure 10.5 because it does not vary with listening mode, and E_{sleep} is omitted because it has a negligible impact on overall energy consumption. The plots for the listening energy consumption in Figure 10.5 illustrate the energy savings of choosing per-node check intervals (See Table 10.1). We observe that ALPL saves more on idle listening energy consumption for nodes at higher levels in the tree, because these nodes have fewer packets to forward and they can listen to the channel less often. For the level 1 node, the idle listening energy consumption is the same for a network-wide listening mode and ALPL because these nodes must listen often enough to accommodate their high forwarding load.

ALPL saves on idle listening energy consumption at the cost of increased transmission and reception energy consumption, which is an inherent tradeoff of the underlying BMAC protocol [56]. The increase in reception and transmission energy consumption stems from the use of longer check intervals, which requires long preambles for packets. Note that the level 1 node has the same transmission and reception energy consumption in both cases because this node receives and sends packets with the same preamble.

Figure 10.6 compares the overall energy consumption for BMAC and ALPL on the basis of node levels. For ALPL, energy consumption follows a similar trend as the idle listening energy consumption in Figure 10.5. For plain BMAC, the overall energy consumption follows the trend of E_r, mainly because E_{listen} is the same for all nodes. Nodes at higher layers exhibit more energy savings with ALPL because their low forwarding load enables them to sleep more often.

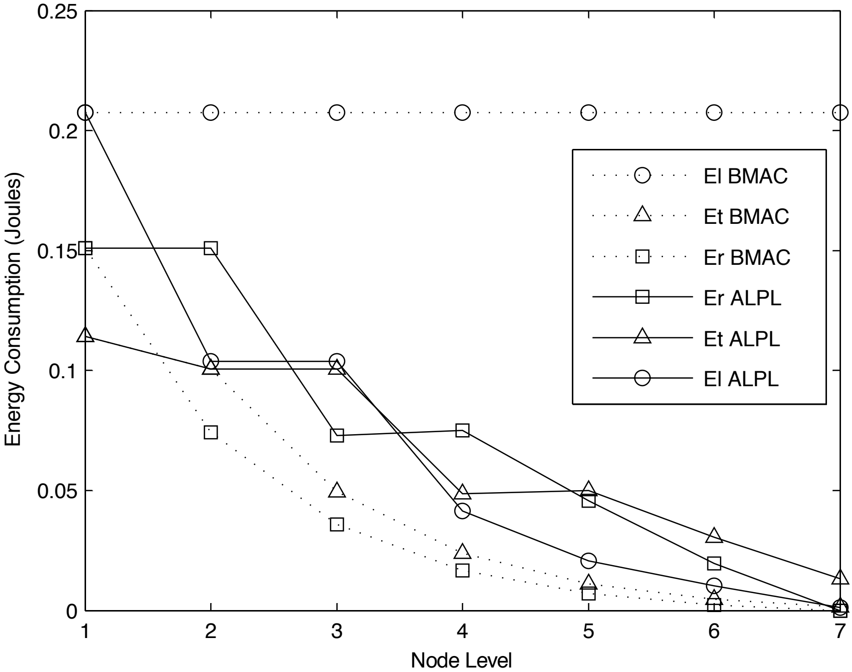


Fig. 10.5. A comparison of BMAC and ALPL for listening, transmission, and reception energy consumption in a sensor network with a binary tree topology

The lifetime of the network is constrained by the energy consumption of level 1 nodes [14, 176]. At first glance, Figure 10.6 seems to indicate that the basic form of ALPL does not extend the lifetime of the network despite significant energy savings at all but one level in the tree. However, recall that sensor networks have a dynamic topology. Varying interference conditions and moving objects typically cause changes in the topology, causing nodes to choose new parents. For instance, a level 2 node may choose the base station as its parent at some point, effectively assuming the role of a level 1 node. Similarly, nodes may choose a parent at a lower level at some point in order to avoid low quality links.

The expanded state representations in Jurdak's framework yield further energy savings at the most loaded nodes which contributes to prolonging network lifetime. By having more detailed information about neighbors' states, nodes can more expressively select neighbors' routing costs. As a result, forwarding traffic moves away from loaded nodes and energy consumption becomes more balanced. In the next 2 subsections, we further describe the interaction of duty cycle and role on energy consumption in the framework.

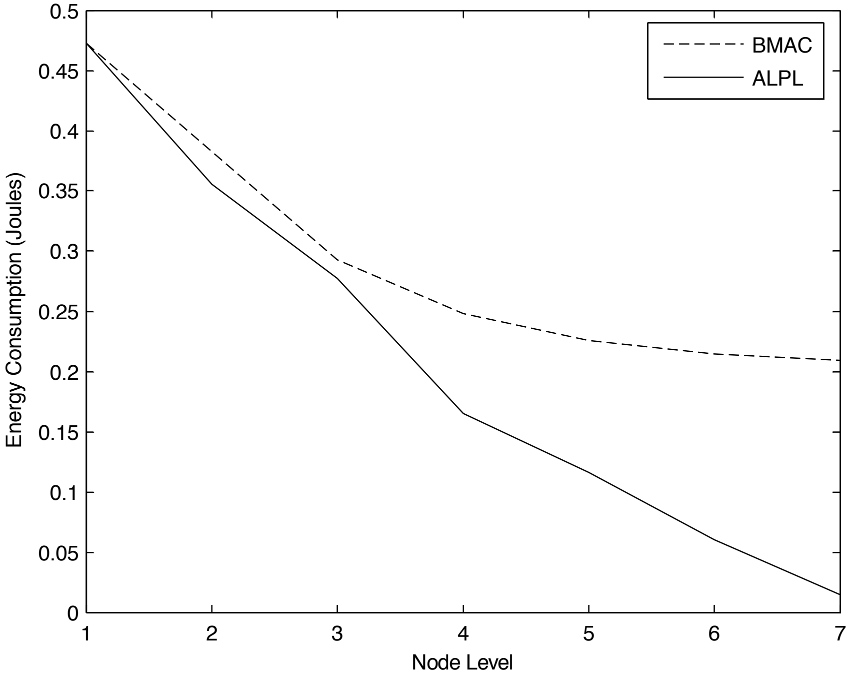


Fig. 10.6. A comparison of BMAC and ALPL for overall energy consumption in a sensor network with a binary tree topology

10.4.3 Duty Cycle

The energy consumption in sensor networks is nonuniform among the nodes due to communication asymmetries. First, data always flows to one or a few sinks. The nodes that are one hop away from a data sink are called critical nodes [176]. Critical nodes have a larger forwarding burden and consume more energy than nodes further away from the sink [14]. These factors indicate that nodes can use up battery resources at different rates. ALPL can contribute to balancing the energy consumption rates among network nodes by manipulating data forwarding patterns and listening modes. Of course, the degree of achievable load balancing depends on the node density and the layout of the nodes.

In terms of Equation 10.6, highly loaded critical nodes have larger E_t and E_r. Because ALPL sets the listening mode according to topology information, a loaded critical node will also choose to listen frequently to the channel, so it has a high E_{listen}. As a result, the energy consumption E and the duty cycle at a loaded critical node are higher than that of its neighbors. Through the optimization framework, the loaded critical node informs its neighbors of its busy state. As a result, the loaded node's children increase its routing cost and choose new parents to forward their packets, thereby reducing its E_r and

E_t at the loaded critical node. The loaded critical node then switches to a listening mode with a longer check interval to reduce E_{listen}. The reductions in E_r, E_t, and E_{listen} cause the overall energy consumption E at the critical node to drop significantly.

10.4.4 Role

In applications where nodes may dynamically select different roles, the sensing energy consumption E_d may become a significant player in determining the node's energy consumption relative to its neighbors. For instance, suppose the occurrence of a particular event causes some nodes in the network to double their sampling frequency in order to better track the event. The increased sampling frequency at these nodes increases the sensing energy consumption E_d and it also increases E_t because the node sends packets more frequently to report the more frequent sensed data. The increase in E_t falls within the radio duty cycle value so it does not cause any distortion for energy balancing. The more appreciable increase in E_d has no explicit effect on the radio duty cycle at the node, but it can cause energy imbalances in the network. Including the sensing energy consumption E_d in routing and MAC decisions remedies this problem and maintains a balance in overall energy consumption E among the nodes.

10.5 Deployment Results

In this section, we investigate the impact of local state-driven optimizations on a test-bed of sensor nodes deployed in our laboratory. The sensor nodes in our experiments consist of 14 mica 2 motes from Crossbow [7]. Our implementation of ALPL is in NesC [199], a component-oriented variant of C customized for networked embedded systems and built into TinyOS.

The nodes are placed at random positions in the laboratory and the base station is placed near one of the walls of the room. We reduce the transmit power of nodes to limit their radio range, enabling multihop communication. The aim of the experiments is twofold: (1) to assess the effect of state-driven optimizations on the global network energy consumption; and (2) to evaluate the local node energy consumption and energy balancing benefits for state-driven optimizations.

We adopt the method suggested by [56] for computing energy consumption. The underlying BMAC design includes several radio states, including active and sleep states. The average current draw for each radio state is fixed. The method employs a counter that keeps track of the time that the radio spends in each power state. Having the current draw and time spent in each state, each node can continually compute its aggregated energy consumption so far. The energy consumption results shown for our deployments represent the aggregated energy consumption for the entire duration of the deployment, including all overhead communication for maintaining routing graphs. To normalize the results, we assign the data point with the highest aggregated energy consumption a value of one, and we assign the remaining values for that comparison according to that data point.

10.5.1 Time-Driven Sensor Network

The first experiment set considers a time-driven monitoring sensor network with a single data sink. We conduct 3 experiments for the same physical network topology. In the first experiment, we initially determine the listening mode for the busiest node in the network, and we assign that listening mode in BMAC to all the nodes. In the second experiment, each node runs ALPL and sets its listening mode according to its number of descendants in the routing tree. The third experiment expands the view of local state to include the duty cycle, so nodes run ALPL and select listening modes according to the expanded local state. We refer to this case as Energy Aware ALPL (EA-ALPL).

Each experiment lasts for 43 hours. In all experiments, nodes run the Surge application that is available with the standard distribution of TinyOS. In Surge, nodes sample the sensors and send the data once every minute. Thus, each node sends 2580 data packets during each experiment. The routing update period for the network-wide listening mode experiment is 120 seconds. For both ALPL and EA-ALPL experiments, the routing update period is 90 seconds, to allow more adaptive link qualities based on routing update messages. The radio duty cycle weight α is set to 2 for this experiment set.

Overall Energy Consumption

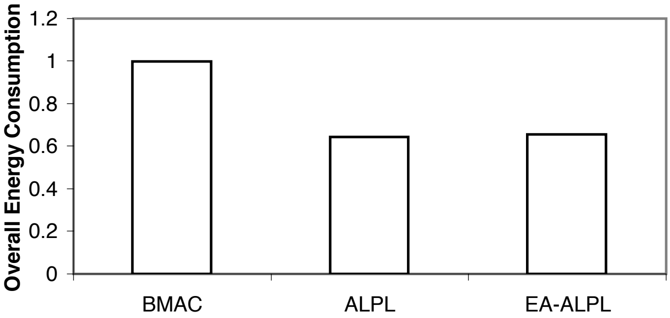


Fig. 10.7. Global network energy consumption during deployment

The average data yield for BMAC, ALPL, and EA-ALPL is about 98.5%. Because the data yield is the same with or without state-driven optimizations, we focus our analysis here on energy consumption issues.

Figure 10.7 plots the average global network energy consumption for BMAC, ALPL, and EA-ALPL. Both ALPL and EA-ALPL reduce global energy consumption by about 35% on average during the deployment. The reduction in global energy consumption stems from the optimal local decisions at each node. In particular, the number of descendants variable is the main contributor for the reductions in energy consumption, as the next subsection reveals in more detail. This reduction also confirms the benefits of greedy local decisions in reducing overall network energy consumption.

The global energy consumption for ALPL and EA-ALPL experiments is almost the same with a slight difference of 1%. The main distinction between ALPL and EA-ALPL is in the distribution of the energy consumption among network nodes rather than in the aggregated energy consumption. The remainder of this subsection sheds more light on the issue of individual node energy consumption that drives the global energy consumption reductions of ALPL and EA-ALPL and the details behind the load balancing benefits of EA-ALPL.

Local Energy Consumption

We now turn our attention to local energy consumption at each node. We collect node statistics by piggybacking node status information into data packet. Each arriving data packet for a node provides one data point regarding node status. Recall that during each 43 hour experiment, each node sends 2580 data packets, yielding 2580 data points for each node.

Figure 10.8 shows the average check interval of each node in the BMAC, ALPL and EA-ALPL deployment experiments, with the error bars indicating the 99% confidence interval. In plain BMAC, all nodes use a check interval of 20 ms. Nodes using ALPL or EA-ALPL can choose one of four check intervals: 20 ms, 50 ms, 100 ms, or 200ms. In both ALPL and EA-ALPL, the average check intervals of all nodes are longer than in BMAC, so all nodes save on idle listening energy consumption.

EA-ALPL yields a more balanced spread of check intervals among nodes than ALPL or BMAC. The nodes with the lowest average check interval in ALPL, such as nodes 1 and 8, have a higher average check interval with EA-ALPL. Similarly, some of the nodes with the highest average check intervals in ALPL have lower check intervals in EA-ALPL. The balancing out of average check intervals yields more balanced energy consumption among network nodes.

The narrow 99% confidence interval for all nodes in the ALPL and EA-ALPL experiments indicates the stability of this approach in the long-term experiments. We also note that busier nodes with the smaller average check interval had slightly less stable check intervals compared to other nodes, due to

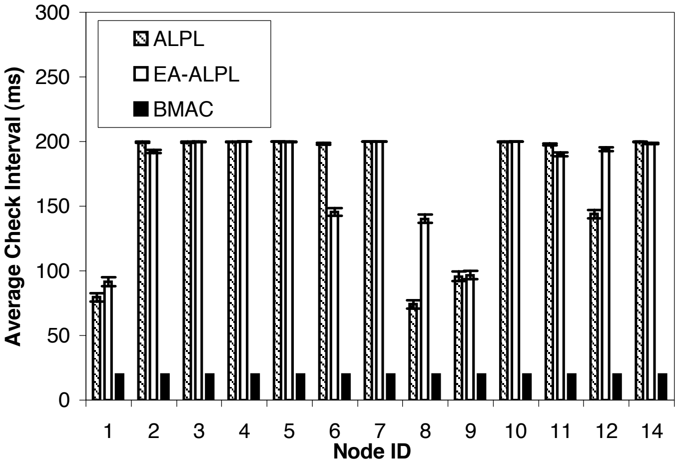


Fig. 10.8. Average check intervals at each node. The error bars indicate the 99% confidence interval

occasional routing oscillations. The stability of check intervals for busy nodes is of the same order for both ALPL and EA-ALPL. Since the basic version of ALPL does not introduce any routing cost modifications, we attribute these transient oscillations to link state changes in the dynamic network topology.

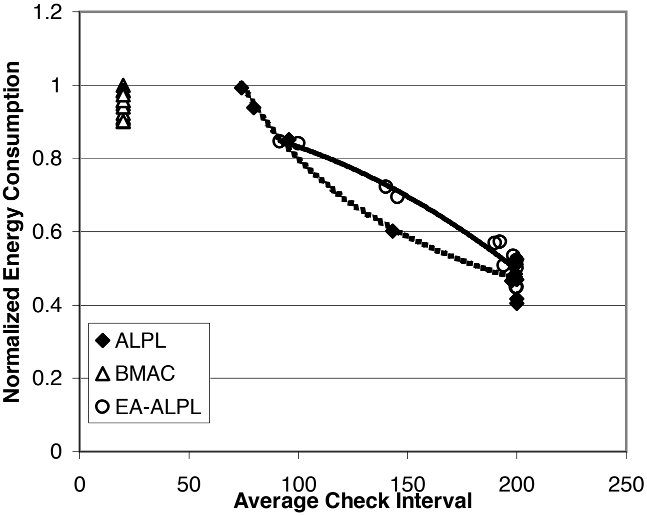


Fig. 10.9. Local energy consumption as a function of average check interval

Figure 10.9 shows the effect of average check intervals on total local energy consumption at each node for the duration of each experiment. Each point in Figure 10.9 relates the aggregated energy consumption of a single node during an experiment to the node's average check interval. In plain BMAC, all nodes consume almost the same energy as the busiest node because all nodes use the same listening mode. In ALPL, nodes with an average check interval close to 200 ms achieve energy savings of more than 50% compared to the busiest node. EA-ALPL yields a more balanced traffic load, as it reduces energy consumption at the most active nodes by about 16% at the cost of small increases in energy consumption at less active nodes. This tradeoff is favorable because network lifetime depends on the most active critical nodes.

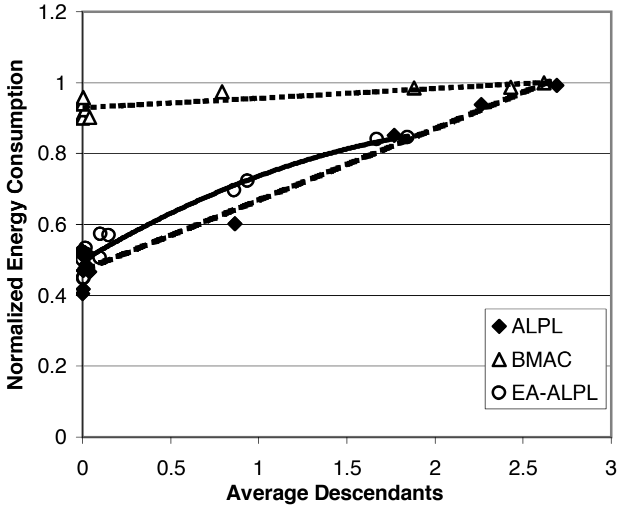


Fig. 10.10. Local energy consumption as a function of the average number of descendants

Through a similar representation to Figure 10.9, Figure 10.10 plots the aggregate local energy consumption of each node based on the node's average number of descendants during the deployment. For both ALPL and BMAC, the range and distribution of the number of descendants is the same because both methods use the same routing metrics. This reinforces the claim in Section 10.3.3. EA-ALPL incorporates radio activity into routing decisions to balance the forwarding load, so the most loaded node in EA-ALPL has an average of 2 descendants in comparison with an average of 2.5 in ALPL and BMAC.

Energy consumption for the BMAC case is correlated with the number of descendants, but the variation is limited. The overall trend for both ALPL experiments is that it yields more energy savings for nodes with fewer descendants, because these nodes can use longer check intervals. As the number of descendants increase, the energy savings of using ALPL are reduced because the average check interval gets closer to the case of network-wide listening modes. EA-ALPL conforms to the trend of larger energy savings for nodes with fewer descendants. However, EA-ALPL yields higher energy savings than ALPL at the nodes with the highest number of descendants, mainly by shifting the forwarding load away from busier nodes when possible. By reducing the energy consumption of the busiest node, the inclusion of radio duty cycle prolongs network lifetime.

10.5.2 Event-Driven Sensor Network

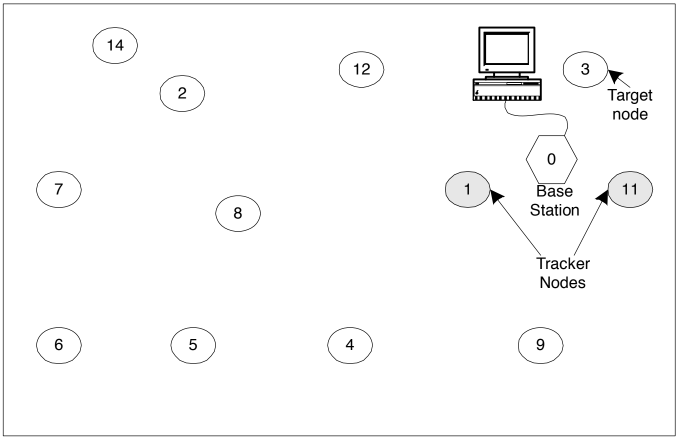


Fig. 10.11. Physical network topology for the event-driven experiments

In this section, we consider an event-driven sensor network application to investigate the benefits of using role information to alter network behavior. Figure 10.11 shows the topology of the test-bed network. The application models a target tracking scenario where the nodes react to the appearance of the target by providing data more frequently. The nodes in the network collect and send their sensor data periodically every 60 seconds by default. We designate one of the nodes, node 3, as the target node. Nodes that detect the target node's presence, nodes 1 and 11 in Figure 10.11, begin sampling their sensors and sending the data at 30 second intervals. Nodes that do not detect the presence of the target node continue sampling their sensors at 60 second intervals.

We conduct 4 experiments for the same physical network topology. The first 3 experiments use BMAC, ALPL, and EA-ALPL as in the time-driven case above. The fourth experiment considers the node's number of descendants, duty cycle, and role as the local states and adapts network behavior accordingly. We refer to this case as Role and Energy Aware ALPL (REA-ALPL). The weights of both the radio duty cycle and sensing cost metrics, α and β are set to 2 for this scenario.

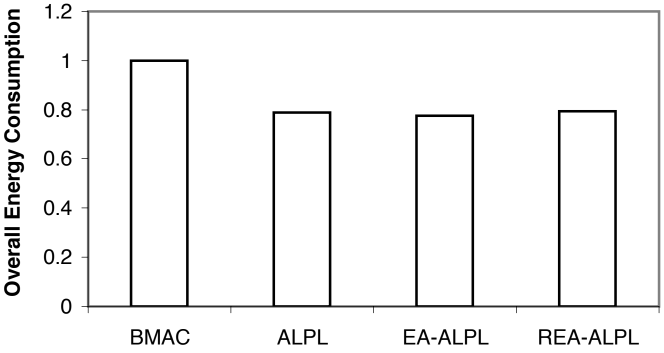


Fig. 10.12. Global network energy consumption during deployment

Global Energy Consumption

The average data yield for BMAC, ALPL, EA-ALPL, and REA-ALPL remains at 98.5%. As in the time-driven case, data delivery rate does not drop when using Jurdak's optimization framework.

Figure 10.12 plots the average global network energy consumption for BMAC, ALPL, EA-ALPL, and REA-ALPL. All three state representations in ALPL yield almost the same overall energy consumption, since they all include the number of descendants variable, which is the main contributor to reduction of energy consumption. ALPL reduces the overall network energy consumption by 21% over the BMAC case for this event-driven network. The reductions in global energy consumption are lower than the time-driven network case because the sensing energy consumption constitutes a larger portion of overall energy consumption in the event-driven case.

Local Energy Consumption

We now turn our attention to the local energy consumption at each node. Before presenting the local energy consumption results, we note that the goal of this analysis is to illustrate how the addition of the role aspect contributes to load balancing, leading to a longer network lifetime. Ideally, all nodes would deplete their batteries at about the same time. In this experiment, the goal is to explore the extent to which nodes can recognize the increased sensing activity of the tracker nodes 1 and 11 in order to avoid these nodes in routing decisions.

We first examine the average check interval for each node during the deployment. Figure 10.13 plots the average check interval of each node for the duration of each of the three ALPL experiments. We omit the constant BMAC check interval from the figure for presentation clarity. In the ALPL experiment, the tracker node with an ID of 1 is the busiest node with a check interval of 43 ms. By including radio duty cycle information, EA-ALPL shifts some of node 1's forwarding load to node 9, enabling a slightly longer average check interval at node 1. REA-ALPL includes information on sensing activity, which shifts most of node 1's forwarding load to nodes 8 and 9. As a result, we observe an average check interval of 183 ms at node 1 in the REA-ALPL, while the check intervals of nodes 8 and 9 drop to 51 and 75 ms respectively. The expanded state representation of REA-ALPL did not cause any added instability in the network, evidenced by the similar width of error bars for the three experiments.

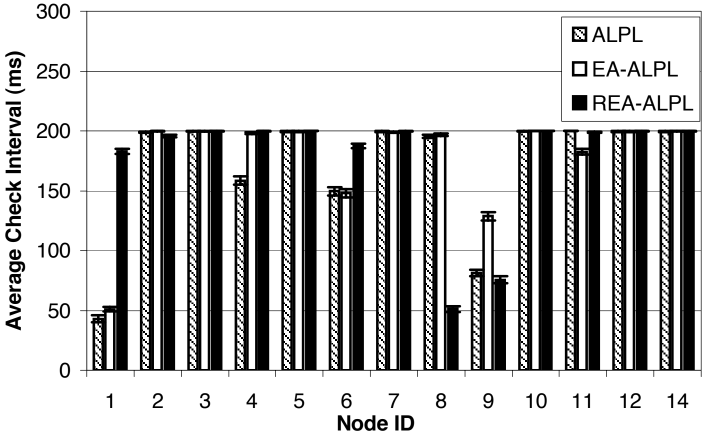


Fig. 10.13. Average check intervals at each node. The error bars indicate the 99% confidence intervals

Figure 10.14 examines the topological distinctions of each experiment by plotting the average number of descendants for each node in the ALPL experiments. Figure 10.14 omits the results for the BMAC experiment results because the number of descendants is the same as ALPL. For the case of ALPL, the top three forwarders have a number of descendants that ranges between 2.62 and 1.1. EA-ALPL considers radio duty cycle as the main reason for energy imbalance, which narrows the gap in the average number of descendants of the three busiest nodes to a range between 1.875 and 1.32. REA-ALPL provides a more expressive node energy profile by considering the potential for increased sensing frequency at tracker nodes. REA-ALPL recognizes that node 1 already has a higher energy burden for tracking the target node and reporting the data at double the frequency of other nodes, so it relieves node 1 from most of its forwarding burden. As a result, REA-ALPL forces nodes to avoid node 1 as a routing parent, which shifts most of the forwarding to nodes 8 and 9. The number of descendants at nodes 8 and 9 experience higher instability than at other nodes in the REA-ALPL experiment, but the absolute instability in the number of descendants for all nodes appears to be small and transient.

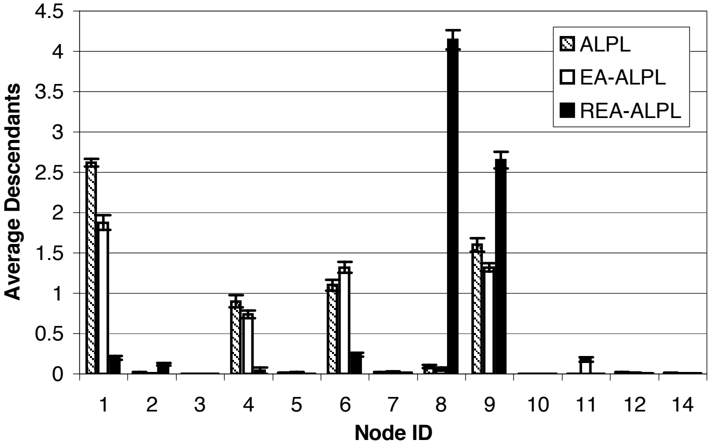


Fig. 10.14. Average number of descendants. The error bars show the 99% confidence intervals

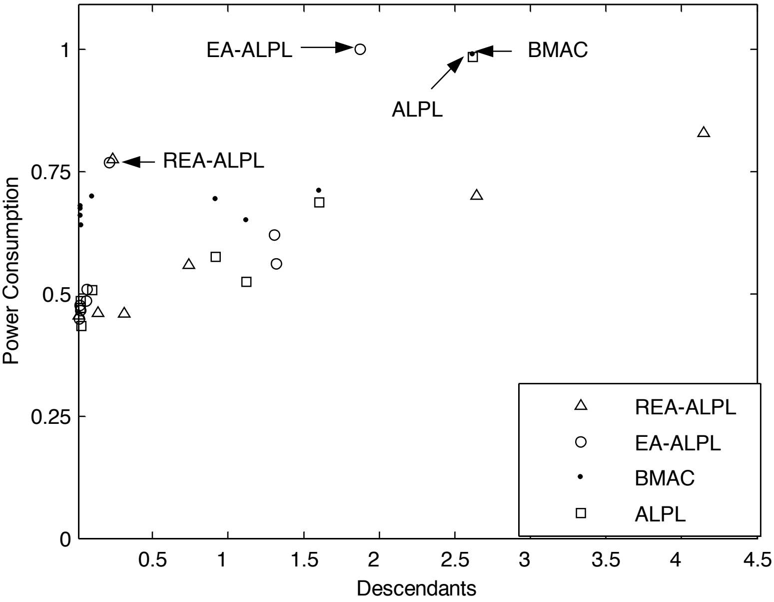


Fig. 10.15. Energy vs. average number of descendants

Figure 10.15 shows the effect of shifting the descendants away from node 1. The labeled arrows in Figure 10.15 indicate the point corresponding to node 1 for the four cases. The energy consumption for node 1 in the cases of BMAC, ALPL, and EA-ALPL is about the same, although EA-ALPL reduces the number of descendants by about 25%. This effect is explained by the dominant effect of sensing energy consumption of node 1 over its listening energy consumption. REA-ALPL reduces the energy consumption of node 1 by about 25%. By shifting the descendants of node 1 to nodes 8 and 9, REA-ALPL enables node 1 to have a longer lifetime than node 8 and keep on reporting the target node even after node 8 dies. REA-ALPL thus provides the most balanced energy consumption of the 4 experiments through an extended cross-layer state representation that considers all the causes of energy consumption imbalance.

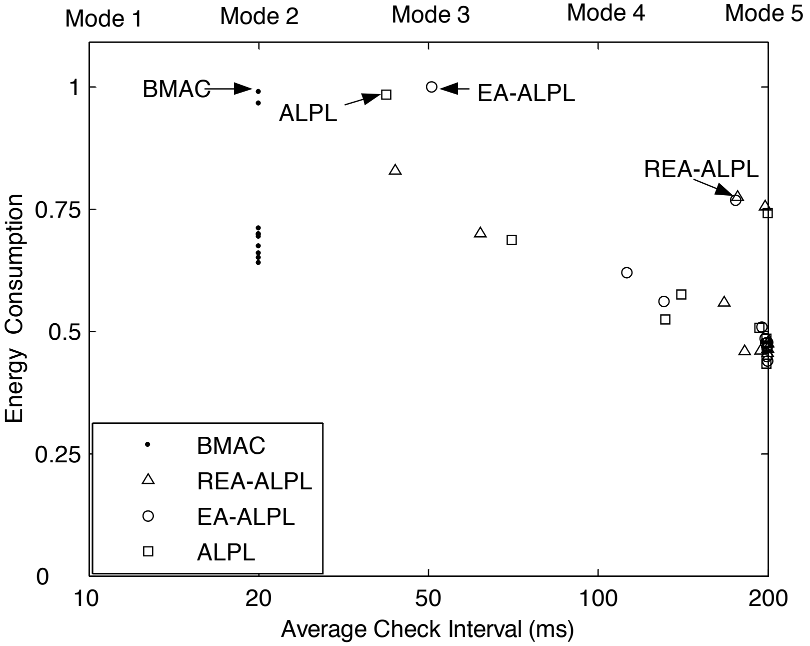


Fig. 10.16. Node Energy Consumption vs. average check interval: the labels at the top of the graph indicate the listening mode index in TinyOS

Shifting descendants away from node 1 reduces both the reception and transmission energy consumption at node 1, but it also reduces the idle listening energy consumption at node 1 as indicated by Figure 10.16. In contrast, the three experiments that do not consider node role fail in reducing node 1's listening energy consumption. Because REA-ALPL diverts almost all forwarding traffic away from node 1, it enables node 1 to have an average check interval of about 183 ms. This increase in check interval significantly lowers the listening energy consumption at node 1, and reduces overall energy consumption at this node. The traffic diverted from node 1 to node 8 causes node 8 to have the shortest average check interval among all 3 state representations. As mentioned before, the traffic shift enables energy consumption balancing in the network. As a result, the plot for REA-ALPL exhibits the lowest variation in the energy consumption of the 4 most loaded nodes for all experiments.

Through its consideration of a more comprehensive definition of node state that includes role information, REA-ALPL has yielded better load balancing in the event-driven tracking network experiments. In particular, it has shifted the traffic load away from the nodes with high sensing activity. This traffic shift has reduced the energy consumption of the busiest node in the network by 23%, resulting in significant improvement of network lifetime and ensuring that the nodes that detect the event remain active for as long as possible.

10.6 Discussion

This chapter has proposed a framework for greedy cross-layer local optimizations in sensor networks that reduces overall energy consumption in the network and promotes load balancing among nodes to customize network behavior to an application's performance requirements. The framework enables nodes to use their local and neighborhood state information to determine their behavior at the MAC layer and at the network layer. At the network layer, a flexible cost function enables nodes to customize routing cost metrics according to an application's performance requirements. At the MAC layer, the nodes set their check intervals in BMAC according to their local state. Bringing together the optimizations at the network and MAC layers, ALPL is a cross-layer mechanism that ensures seamless adaptation to local state.

We have validated the framework through two sets of experiments on a test-bed of sensor nodes. The first set of experiments represented a typical time-driven monitoring sensor network with a single data sink. The second set of experiments was an event-driven network that modeled a target-tracking application.

The experiments have explored three representations for local node state. The analysis and results show that including more information from across the network stack into the local state representation better reflects nodes' energy profiles and enables more informed adaptations of network behavior. In the time-driven case, the state representation combining duty cycle and descendants yields the most balanced energy consumption distribution. In the event-driven case, the state representation that includes role, duty cycle, and descendants yields the most balanced spread of energy consumption for the event-driven case.

The adaptive and flexible nature of Jurdak's framework supports the dynamic nature of sensor networks and can exploit local state information about the present, the past, and the predicted future state of sensor nodes to reduce energy consumption. We have studied how adapting listening modes to the node's current logical topology position (which represents the node's present state), duty cycle (which represents the node's past state), and role (which represents the node's predicted future state) can reduce the global network energy consumption and the local energy consumption at each node.

The use of a proactive routing strategy for sensor networks requires careful tuning of the routing update period for the network application in order to balance route adaptivity and energy efficiency. In long term monitoring applications, sending route update message with a period of the same order of data message periods ensures that the communication overhead of proactive routing is small.

One concern of using greedy approaches is that local decisions may not be globally optimal. For instance, a well-known case for greedy approaches is when the cheapest next hop does not represent the best path to the destination. In Jurdak's framework's implementation, the shortest path algorithm in MintRoute uses a modified cost metric including link quality and energy to route packets to the next hop. Because each node learns its hop count to the base station by adding a single hop to its parent hop count, nodes are guaranteed to have an accurate view of their hop count. In contrast, nodes learn the link quality and energy cost metrics for their direct neighbors only, so nodes may not adapt instantaneously to abrupt changes for these metrics elsewhere in the network. However, we have observed through our deployment that the nodes' energy cost metrics change gradually rather than abruptly. As for link qualities, transient changes occur due to movement of objects around the nodes, but the high data yield rate in our deployment has shown that these changes have minimal effect on data delivery or energy savings.

Jurdak's customized framework adopts BMAC for the MAC layer protocols and introduces ALPL to interface the MAC layer with a proactive strategy at the routing layer. For any proactive routing protocol, ALPL does not significantly increase communication, processing, or storage complexity. The results in Section 10.5 have shown that the communication overhead of a few bits of state information in periodic routing messages is a small price to pay for the energy savings of ALPL. In terms of processing complexity, cost metric calculations in ALPL involve simple arithmetic operations and routing decisions involve a few simple conditional statements. Finally, the state information stored at each node is a function of node density and not network size, which adds scalability to the greedy approach.

Because of the reduced transmission power in our deployment and the limited space in the laboratory, each node had an average of about 4 neighbors, yielding a relatively high network density. The high node density in our experiments raised the degree of contention among nodes. Coupled with the framework's locally optimal decisions, the experiments have confirmed that the framework is scalable to large or dense networks despite the limited size of our network test-bed.

ALPL attempts to address the cross-layer design challenges discussed in Section 7.3.3. ALPL's design advocates long-term proliferation as it provides a modular framework for new services to enable optimization different network parameters. The framework in this chapter adheres to the guidelines of the SP architecture [149] that designates the "thin waist" of a sensor network architecture to be between the MAC and the routing layers. ALPL also provides a holistic perspective of performance optimization, enabling algorithms at different layers to influence the listening mode. Finally, ALPL clearly identifies the interactions with current protocols, mainly the routing and the MAC protocols, and it outlines the required configurations for maintaining sound network operation.

Reduction of global network energy consumption through local decisions is an approach that is widely applicable to many sensor network applications and quality-of-service requirements. Jurdak's framework is independent of the underlying routing protocol or MAC protocol. Instead, it can build on other underlying mechanisms with a modular design to optimize the behavior of other ad hoc and sensor network applications.

11

UWB Ad Hoc Network

This chapter customizes Jurdak's optimization framework for UWB ad hoc networks in order to promote 4 performance goals: (1) maximizing throughput; (2) promoting fair access; (3) minimizing connection setup latency; and (4) minimizing control overhead. Figure 11.1 shows the customized framework to achieve the global policy for an UWB network. Each node maintains a local state table that includes perceived interference, received radio power, communication resource allocation for different traffic classes, and reliability. Nodes use this state information to adapt the transmission rate and power of their active links in order to maintain link quality guarantees. Nodes also adapt the neighbor costs and the period of their state declaration (hello) messages according to changes in local and neighborhood state.

Within the optimization framework, we present U-MAC [200], an adaptive Medium Access Control (MAC) protocol for UWB in which nodes periodically declare their current state, so that neighbors can proactively assign power and rate values for new links locally in order to optimize global network performance. Simulations comparing U-MAC to the reactive approach confirm that U-MAC lowers link setup latency and control overhead, doubles the throughput and adapts better to high network loads. Simulations also reveal that the basic form of U-MAC favors nodes that are closer to the receiver. As a result, we also introduce novel mechanisms that control the radius around a receiver within which nodes can have fair access to it. We show through simulations the effect of the mechanisms on the tradeoff between network throughput and fair access.

Portions reprinted, with permission, from (R. Jurdak, P. Baldi, and C. V. Lopes. "U-MAC: A Proactive and Adaptive UWB Medium Access Control Protocol". Wiley Wireless Communications and Mobile Computing Journal, 5(5):551-566, 2005).

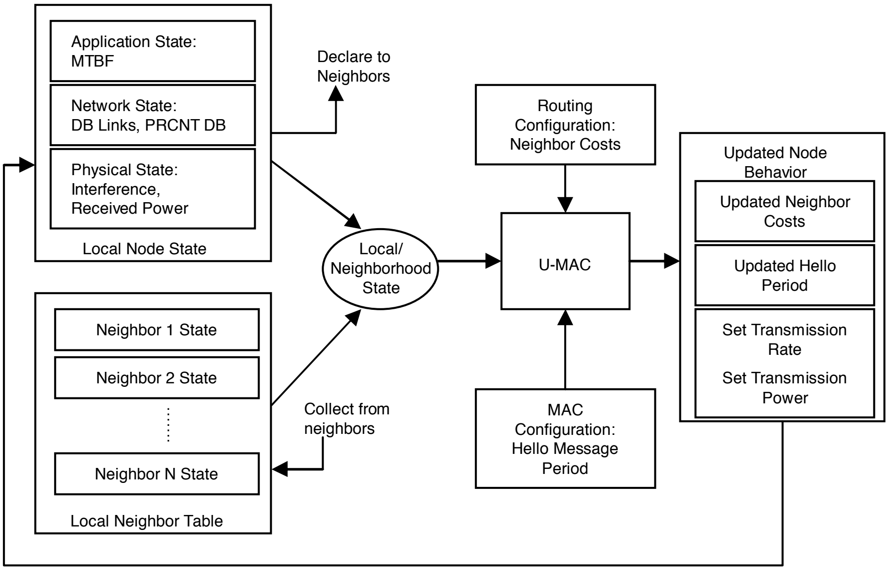


Fig. 11.1. Customized framework for UWB ad hoc network

11.1 Introduction

Ultra sideband radio (UWB) is a spread-spectrum technique that is based on the modulation of short nanosecond pulses. The short duration of pulses results in the thin spreading of the signal over a large spectral bandwidth. Consequently, UWB communication is robust to frequency selective and multipath fading and supports high data rates. In recent years, UWB has received increasing recognition for its applicability to short to medium range communication networks [201, 202] because of desirable features such as high data rates, low power consumption, precise ranging capabilities, resistance to multi-path fading, and penetration of dense objects. UWB is currently a candidate technology for short range high transfer rate applications such as the simultaneous transfer of multiple video streams in a Wireless Personal Area Network (WPAN) [203]. It is also being considered for medium range sensor networks with lower transfer rates [204].

UWB also has an established positioning capability because it has been used for ground-penetrating radar applications by the military for more than half a century. UWB's positioning capability makes it suitable for sensor networks in the presence of physical obstacles, such as walls. Similarly, UWB is suitable for wireless communication in sensor networks that are embedded in the ground, for instance monitoring the soil in agricultural fields, or in man-made physical structures, such as bridges.

A central problem in UWB networks is the joint optimization of transmission power and transmission rates for active links. The joint rate and power assignment problem in UWB involves complex tradeoffs between fair rate assignment, network efficiency, and QoS, which cut across traditional layer boundaries. A high power link may achieve high transmission rates, but it also causes high interference which limits the rate available to neighboring links. On the other hand, a low power link promotes fair access to the wireless medium but yields lower transmission rates. Thus, nodes must collaboratively determine the optimal rate and power values for new links in the network.

Another important consideration for rate and power assignments in UWB networks is fair access among nodes at different distances from a common receiver, especially in *ad hoc* networks with dynamic topologies. An inherent property of wireless communications is that transmission rates drop with increasing distance. This effect is even more pronounced in UWB communications, where the strong correlation of transmission rates with multiuser interference levels further increases the impact of relative distances on transmission rates. This impact may be unsuitable for applications that require all the network nodes to have fair access to the medium regardless of distance. The fair access requirement in such applications imposes an additional constraint on the choice of transmission power.

So far, UWB research has been primarily confined to the investigation of the behavior of the physical layer [10, 205]. Research at higher layers of the network stack has been somewhat limited. Previous research proposals for higher layers considered an underlying Radio Frequency (RF) physical layer, so most of these proposals are not suitable for UWB networks. For instance, the existing wireless MAC protocols [179] for RF networks do not meet the need of UWB networks for joint rate and power assignment, hence the need for new UWB MAC protocols. There have been recent attempts to develop mechanisms for UWB networks at the MAC layer. The work in [206] proposes a simple reactive multiple access protocol that defines the handshaking procedure to establish a new link. The work in [207, 208] discusses a protocol that uses periodic state updates to allow nodes to jointly assign rate and power assignment values locally at each node.

Here, we examine the U-MAC (Ultra sideband MAC) protocol that jointly assigns rate and power values in UWB networks, and reduces the control messaging and latency required for link establishment. To ensure collaboration among nodes, U-MAC requires nodes to periodically announce their state information (from the application, network, MAC, and physical layers) in hello messages, so that any node can locally select appropriate rate and power values for a link request without polling neighbors. In U-MAC, the hello message period adapts to the stability of network state, to avoid sending frequent updates unnecessarily. U-MAC also provides a mechanism to adjust the radius around a receiver within which all nodes get fair access to the receiver. Furthermore, the protocol's framework supports the future integration of multihop links [175], which limits the impact of distance on fairness and the internode interference.

Within the emerging UWB MAC framework, the main contributions in this chapter are:

- The introduction of adaptive periods for hello messages in an UWB network so that control message overhead is minimized
- A comparative assessment with the reactive approach regarding control overhead, link setup latency, network throughput and adaptability
- The development of mechanisms that promote fair access among nodes in an UWB network

The remainder of the chapter is organized as follows. Section 11.2 reviews the fundamental concepts in UWB radio and provides the framework upon which we design U-MAC. Section 11.3 introduces U-MAC and explains the mechanisms and features that characterize this protocol. Section 11.4 presents the simulation results for U-MAC. Section 11.5 discusses U-MAC in light of existing literature and future research directions.

11.2 UWB Network Principles

11.2.1 UWB Principles

Recently, UWB radio has received increasing recognition for its applicability to multiuser wireless communication networks. UWB radio relies on periodic sequences of short subnanosecond pulses (referred to as monocycles) for data transmission. The short duration of UWB pulses yields a low power spectrally wide signal. In a single sender/receiver environment, a common modulation technique used with UWB radio is Pulse Position Modulation (PPM), which encodes symbols by shifting the monocycles according to the following expression:

$$s(t) = \sum_{i=-\infty}^{\infty} \sum_{j=0}^{N_S-1} g(t - jT_f - b_i\tau) \tag{11.1}$$

where $s(t)$ is the transmitted signal, $g(t)$ is the pulse, and T_f is the frame time. N_s is the number of pulses that encode each symbol, and the sequence b_i encodes the information bits. Win and Scholtz [205] have proposed a multiuser access scheme for UWB using time-hopping (TH) codes. TH-codes accommodate multiple users by further shifting the pulse $g(t)$ according to one of many chipping codes. Consequently, UWB radio has potential for supporting multiple users within the same frequency and spatial channel.

In a centralized UWB network, the base station could send periodic beacons to allow nodes to stay synchronized. Global synchronization among nodes in an *ad hoc* UWB network requires excessive signalling overhead, which is a waste of valuable battery resources in mobile nodes. Consequently, it is more realistic to assume that only each sender and receiver that share a link are synchronized. This assumption may lead to collisions of some monocycles among different links.

The impact of monocycle collisions due to the lack of global synchronization is reduced by sending multiple pulses for each symbol at the source as a form of forward error correction, so that collisions contribute only to mutual interference. Provided that proper quality margins are set, collisions only reduce signal quality and do not affect correct reception of data at the receiver[1].

The binary bit rate of an M-ary PPM UWB signal is given by the following expression [209]:

$$R_b = \frac{1}{T_f N_S} log_2 M \tag{11.2}$$

Both T_f and M are difficult to modify in an UWB system. Changing M for different transmissions is undesirable for communication systems since it leads to processing overhead. Similarly, modifying T_f for each transmission increases the complexity of the hardware design of the system. Thus, the simplest way to adjust R_b is to vary the value of N_S. The only requirement for allowing different N_S values is that the receiver of each link must integrate the correct number of pulses for each symbol received on that link. A protocol that is adaptive to network behavior should vary N_S based on interference levels in the network. More specifically, high interference levels increase the probability of pulse collisions, which requires more pulses per symbol.

We adopt the framework of Cuomo et al. [206] for an UWB radio resource sharing model that assumes continuous values for R_b. The framework considers that a new link request arrives when there are N pairs of communicating UWB terminals, with each pair consisting of one transmitter and one receiver. Each pair of sender and receiver are synchronized to the TH-code of their common link, and both background and UWB noise impact the SNR of UWB links. Consequently, the SNR at the receiver of the i^{th} link is:

$$SNR_i = \frac{P_i}{R_i PL_{ii}(\eta_i + T_f \sigma^2 \sum_{k=1,k\neq i}^{N} P_k g_{ki})} \tag{11.3}$$

where P_i is the power of the i^{th} transmitter, R_i is the binary bit rate of the i^{th} link, η_i is the background noise energy plus interference from non-UWB sources at receiver i, PL_{ij} is the path loss from the i^{th} transmitter to the j^{th} receiver, g_{ki} is the path gain from the k^{th} transmitter to the i^{th} receiver, and σ^2 is a parameter depending on the shape of the monocycle. Common values for the above parameters are [205]: $T_f = 100ns; \sigma^2 = 1.9966 \times 10^{-3}$; and $\eta = 2.568 \times 10^{-21} V^2 s$; with a pulse duration of 0.75ns.

11.2.2 UWB Traffic Classes

We consider two traffic classes for UWB networks, in accordance with the specifications of the European Whyless Project [206, 210, 211], to address the

---

[1] This concept is similar to increasing the processing gain in CDMA systems.

requirements of different application types: (1) Reserved Bandwidth (RB); and (2) Dynamic Bandwidth (DB).

The RB traffic class is geared towards continuous, real-time or multimedia traffic, since it requires quality guarantees prior to establishing a link. The continuous nature of traffic that exploits the RB traffic class requires that the link rate remain constant throughout the lifetime of the link.

DB traffic does not offer any rate guarantees and is thus suitable for best-effort data, such as internet traffic. As the name implies, the rate of a link is dynamic and elastic, and depends on the number of other active DB links and on interference levels in the network. For instance, if the traffic load in a network is low, then individual DB links may use higher rates.

In short, the goal of RB traffic is to offer a certain quality-of-service for the sender under varying network conditions. The goal of DB traffic is to provide adaptive and efficient overall network behavior for asynchronous data, and to ensure a constant interference level of bursty traffic by modifying rates of all DB channels dynamically.

11.3 U-MAC Protocol

This section presents the design details of the U-MAC protocol. Section 11.3.1 defines the joint power and rate assignment problem in UWB networks. Section 11.3.2 provides an overview of the U-MAC protocol. Section 11.3.3 discusses the topology consideration in U-MAC. Section 11.3.4 covers the periodic internode interactions in U-MAC, namely the periodic hello messages. Section 11.3.5 proceeds to explore the interlayer interactions within a node for determining the transmission rate and power for a new link locally, before proceeding with a link establishment request. Finally, Section 11.3.6 focuses on the tunable MSI margin in U-MAC, which controls communication radii and fairness.

11.3.1 Problem Definition

U-MAC addresses the joint rate and power assignment problem for UWB links for both RB and DB traffic. In general, each node in the network is the receiver for a certain number of communication links. Based on the quality requirements of its currently active links, the node can tolerate a finite amount of additional interference, referred to as maximum sustainable interference (MSI) [212]. The MSI at each node must be efficiently and fairly divided up between RB and DB traffic and among links of each traffic class.

Initially, each node's resources are split evenly between RB and DB traffic. As a node starts receiving link requests, the MSI portion allocated to each traffic type can adapt to the relative number of links in each traffic class. In general, at any point in time, each node allocates a portion λ of its MSI to DB traffic, and the remaining portion $(1 - \lambda)$ to RB traffic, where λ is less than one. Section 11.3.6 provides a more detailed discussion of an MSI allocation technique that avoids starvation.

From the transmitting node's perspective, the challenge is selecting rates and power levels for new links that adhere to the MSI states of it's neighbors. The generic relation between link quality, transmission rate, and transmission power is:

$$Quality\ \alpha\ (\frac{Power}{Rate \times Noise}) \tag{11.4}$$

During the lifetime of the link, new communication links may cause the noise to increase, which subsequently causes quality degradation of the link. Avoiding quality degradation can be achieved in several ways:

1. Increasing transmission power
2. Decreasing transmission rate
3. Providing a quality margin above the minimum quality requirement initially so that when new links are set up, the link can tolerate additional interference

Although increasing the transmission power maintains link quality and transmission rate, it degrades the quality of neighboring links which may require additional power or rate adjustments in the network. Furthermore, the FCC has imposed tight limits on UWB emissions [213], so increasing transmission power to maintain quality is impractical. Alternatively, the link transmission rates can be lowered to maintain quality in response to increasing interference. This option does not require reconfiguration of neighboring links, but it leads to inefficient use of the medium and may cause quality violations if the link carries RB traffic. Thus, U-MAC allows reducing link transmission rates only for DB traffic. Finally, providing quality margins avoids both rate and power adjustments of any active links, which suits RB links. The drawback of quality margins is that they also lead to less than optimal medium utilization.

U-MAC adopts Signal to Noise Ratio (SNR) as the main link signal quality metric. The SNR of a new link must have some margin above the minimum acceptable SNR for the link. In U-MAC, the parameter μ determines the size of the SNR quality margin of links (see (11.7) and (11.9) below). The value of μ could be static or adaptive to the spatial distribution of nodes, the traffic load, and the lifetime of the link. The rest of the discussion assumes that source nodes set the SNR margin μ statically for simplicity.

11.3.2 Protocol Overview

The U-MAC discussion in this chapter covers both RB and DB traffic for a single hop distributed topology. The discussion for a distributed topology could easily apply to the centralized or hybrid topology, since the distributed case is inherently more complex in nature. Furthermore, a single hop topology could be extended to a multihop topology through the use of a global cost function to enable multihop links [175].

The main design goals of U-MAC are to jointly optimize rate and power values in the network to achieve fairness, maximize throughput, and minimize latency and control overhead. To achieve these goals, U-MAC adopts a proactive approach in reporting cross-layer state information.

Rate and power assignments in U-MAC occur at the source prior to sending any control messages. To enable local assignments at the sender, all nodes periodically update their neighbors with their state information through hello messages. Because frequent hello messages may increase interference in the network, hello message periods always adapt to the stability of network state (see Section 11.3.4). Thus, a highly stable node sends hello messages rarely, while a highly dynamic node frequently updates its neighbors about its state. Every node collects and stores each of its neighbors' most recently advertised state information. Significant state changes at a node also trigger hello messages. Triggered hello messages ensure that each node has a sufficiently up-to-date view of the state of its neighbors.

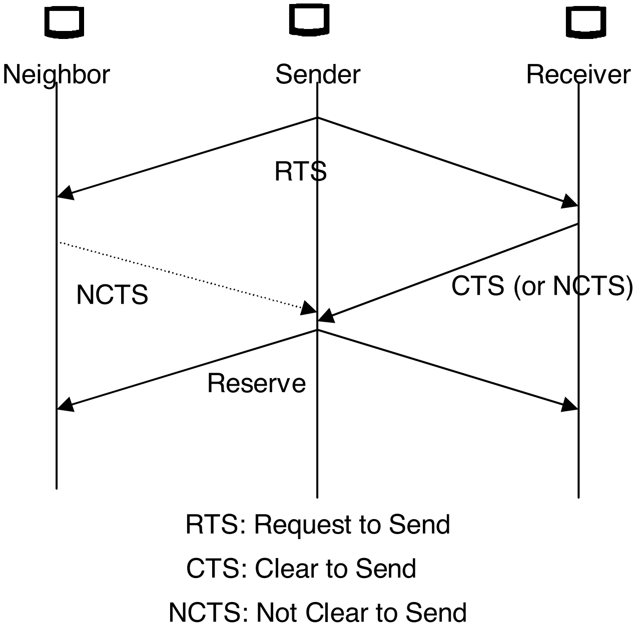


Fig. 11.2. Control message exchanges

Figure 11.2 illustrates the control message exchanges in U-MAC. To set up a new link, a sender S first sends a link request in a Request To Send (RTS) message indicating the rate and power values to the intended receiver R. Upon receiving an RTS, node R and all other neighbors of S check whether the requested link is admissible. If so, then R notifies S with a Clear to Send (CTS) control message, while other neighbors of S refrain from sending any replies if the link parameters are satisfactory. However, if the receiver node R or any other neighbor of S does not agree with the parameters of the new link, then that neighbor notifies S with a Not Clear to Send Message (NCTS) that it should reduce either the transmit power or rate or both. After S collects all the replies, it declares the duration and parameters of the new link (which may have changed according to neighbor replies) in a Reserve message, and immediately sets up the link.

11.3.3 Topology

U-MAC supports a hybrid multihop topology, which provides a node with flexibility in switching between centralized mode when an access point is available, or *ad hoc* mode when an access point is not reachable. To determine its current mode of operation, a node monitors a dedicated hello message channel. Whenever it detects any access point hello messages, it switches to centralized mode. The node must hear access point hello messages periodically; otherwise, it switches to distributed mode. The remainder of the discussion in this section focuses on the case of distributed mode in a single hop topology. U-MAC can easily be extended to support multihop links through a global cost function that quantifies link costs in order to determine optimal routes [175].

11.3.4 Hello Messages

U-MAC requires nodes to advertise their local states periodically through hello messages [214], which provides for quick and appropriate rate and power assignments in the network. Note that although hello messages are periodic for one node, they are asynchronous among different nodes in the network, which helps avoid hello message interference from different nodes. Many factors contribute to avoiding several simultaneous transmissions of hello messages:

- Because the hello message period in each node depends on the node's stability, hello periods are not the same across nodes
- Clock skews contribute favorably to collision avoidance in hello messages of nodes that have the same hello message period and that enter the network at the same time
- The transmission time of hello messages is much shorter than typical hello message periods which further reduces the chances of collisions

Hello messages implicitly provide nodes with ranging information about neighbors, and they explicitly advertise important local parameters to neighboring nodes. Storing recent neighbor state information locally enables a node to make decisions on rate and power assignments for new links, and to make routing decisions for multihop links.

First, nodes use hello messages to determine distances of neighboring nodes. Each node sends its hello message at a fixed power level know a priori to all nodes. Whenever a node receives a hello message from a neighbor, it can estimate the current distance of that neighbor by examining the received signal strength of the hello message and by applying the appropriate propagation model. The current distance from a neighbor enables a node to compute the path loss to that neighbor locally.

Format

Node ID	MTBF	PRCNT DB	DB Links	MSI	Current Interfer.

Fig. 11.3. Hello message format in distributed mode

In addition to providing ranging information, hello messages advertise local state information to neighbors. Figure 11.3 shows the format of hello messages in distributed mode.

Mean Time Between Failures (MTBF) is a measure of a node's communication reliability [175], which is an application layer attribute of a node. The next two fields in Fig. 11.3 are network layer parameters that pertain to DB traffic. The "PRCNT DB" field in a hello message holds the parameter λ, which we introduced in Section 11.3.1. In the "DB links" field of a hello message, a node indicates the number of its active DB links. This field, along with "PRCNT DB" enables a neighbor with bursty traffic to choose a fair rate and power level for a DB link. The last two fields in Fig. 11.3 are physical layer variables that are common to both traffic types. Maximum Sustainable Interference (MSI) information in a hello message presents a node with an upper bound of the tolerable interference at a neighbor [212]. Finally, each node also advertises the aggregate received power of all the active links in its range. This field provides neighboring nodes with recent interference levels, which is useful for selecting rates and power values locally.

Each node compiles information contained in incoming hello messages into a small neighbor table, and the node clears a neighbor entry in the table when it no longer detects the neighbor's hello messages. The storage capacity for the neighbor table is not a major issue for current memory technology.

Period

Because state changes in nodes occur with varying frequency, the hello message period at each node should adapt to the frequency of the node's state changes. More specifically, the period of hello messages should increase with increasing node stability so that unnecessary hello messages are avoided. Node stability combines the effects of the node's mobility, its physical reliability, and its degree of state changes. To quantify the first two parameters, we assume that each node can estimate its velocity and its communication reliability (MTBF), which account for positional and physical stability respectively. A node is physically reliable if its hardware and software components are robust and do not experience frequent interruptions in service. A node has positional and communication reliability if its velocity relative to its neighbors is small. Baldi et al. [175] provide a combined measure of these two factors: $C(\mathit{quality})$.

To monitor the stability of its interference state, each node can compare its current state to the state it advertised in its last hello message. Significant changes in MSI or current interference levels trigger the node to send an early hello message and lower its hello message period. The creation of a new link could result in state changes at more than one node and trigger them to issue hello messages. Consequently, nodes that detect local state changes upon the creation of a new link must wait a random time (within a maximum wait time) before sending a hello message in order to minimize interference on the hello message TH-code.

We define a new Boolean cost metric $C(\mathit{state})$ which takes the value of one if either MSI or the current interference at a node vary beyond their respective thresholds, and takes the value of zero otherwise. When $C(\mathit{state})$ has a value of one, a hello message is triggered. We also define a compound cost metric:

$$C(\mathit{stability}) = C(\mathit{quality}) + C(\mathit{state}) \tag{11.5}$$

which represents the overall stability of a node. U-MAC varies the hello message period at each node linearly with $C(\mathit{stability})$ at that node, as indicated below:

$$T_{hello} = \begin{cases} T_{min} & C(\mathit{stability}) \geq C_{max} \\ K + T_{max} & C_{min} < C(\mathit{stability}) < C_{max} \\ T_{max} & C(\mathit{stability}) \leq C_{min} \end{cases}$$

$$K = \frac{T_{min} - T_{max}}{C_{max} - C_{min}} \times C(\mathit{stability})$$

where T_{max} and T_{min} are the maximum and minimum time between hello messages respectively, and C_{max} and C_{min} represent the upper and lower bounds of $C(\mathit{stability})$ respectively. Finally, nodes that do not experience state changes between two consecutive hello messages increase their hello message period by 1 second as long as the period is lower than T_{max}.

11.3.5 Rate and Power Assignment

In a centralized UWB network, the access point determines optimal rate and power assignments. Nodes that are out of range of an access point and nodes in UWB ad hoc networks must assign channel rates and transmit power levels in a distributed way. Choosing an appropriate channel rate and power is not simple, since nodes do not have a global view of network state. Each node can use its neighbor state information to select appropriate parameters for a new link request.

Most conventional wireless networks that use multiple channels require an explicit channel separation mechanism, such as TDMA or FDMA, to accommodate multiple users [179]. In UWB networks, each pseudo-random time-hopping (TH) code constitutes a separate channel.

In U-MAC, all nodes use a known TH-code as a common control channel. We also assign another fixed TH-code to a dedicated channel for hello messages. Occasional hello message losses are not as critical as control message losses. We assume that nodes synchronize prior to sending and receiving hello messages and control messages. To achieve on-demand synchronization on these two channels, one node could send a short beacon prior to sending its control or hello message to allow neighbors to synchronize to its transmission, which is similar to IEEE 802.11 [5] synchronization for the distributed case. Finally, each of the remaining TH-codes is a potential one-way separate data channel. Synchronization on data channels is only required between each sender and receiver pair of an active link.

RB traffic

The RB traffic class accommodates data streams that require a particular quality-of-service. The two quality parameters of interest are the link transmission rate and the SNR at the receiver of the link. Providing a link rate guarantee often prevents any adjustment of the transmit power level while the link is active, in order to maintain the signal integrity at the receiver. Thus, the goal is locally assigning link rate and power values that make efficient use of the medium, achieve fairness among nodes, and ensure that the quality guarantees (transmission rate and minimum SNR) can be maintained for the lifetime of the link.

First, a node S determines the maximum allowable transmit power level for all neighbors, using the following equation [206]:

$$P_{allowed} = min\{\frac{(1-\lambda_i)MSI_i \times PL_{si}}{T_f \sigma^2}\} \tag{11.6}$$

where MSI_i is the MSI value announced by the i^{th} node, and PL_{si} is the path loss from node S to neighbor i. In short, node S must select a power level that does not violate the interference threshold of any active links at its neighbors. If there are no active links in the network, then $P_{allowed}$ takes the value of P_{max}.

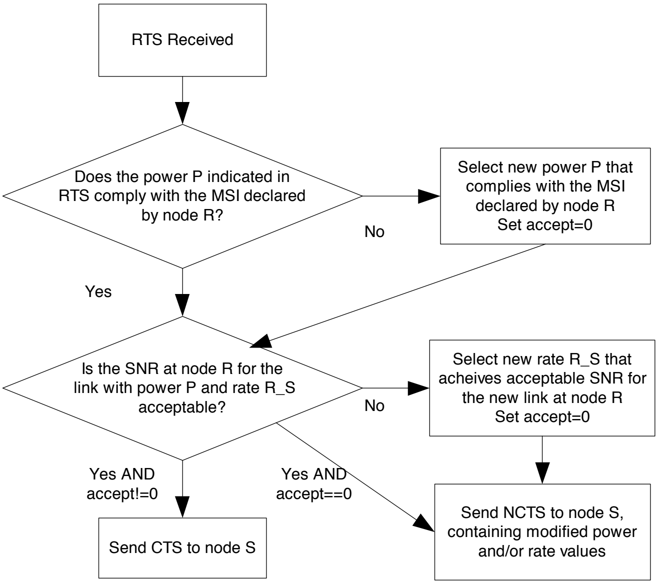
Next, node S must select an appropriate rate for the new link to the receiver R, through the expression:

$$R_S = \frac{min(P_{allowed}, P_{max})/PL_{sr}}{SNR_{min} \times \mu(\eta_r + U_r)} \tag{11.7}$$

where P_{max} is the maximum allowable emitted power from an UWB transmitter, U_r is the combined received power level at the intended receiver, and η_r is the thermal noise level.

R_S should also meet the QoS requirements of the higher layers at node S. Suppose that the network layer at node S requested a desired rate R_{QoS} and minimum acceptable rate R_{Min}. If the value of R_S from (11.7) is higher than R_{QoS}, then R_S is set to R_{QoS}. On the other hand, if R_S is lower than R_{Min}, then node S rejects the request at the MAC layer.

After computing R_S locally, S selects a random TH-code (other than the control channel and the hello channel TH-codes) and initiates a sequence of control messages. If S does not detect that any of its neighbors is attempting to set up a new link, S sends a Request to Send (RTS) message containing the TH-code, $min(P_{allowed}, P_{max})$, R_S, and R_{Min} on the common control channel. S then listens for any replies from its neighbors on that channel. The purpose of the RTS message is to ensure that link requests are serialized and that the establishment of this link is recorded and approved by the neighbors of node S. Because S had selected R_S based on its recent local view of the network, all neighbors of S accept the transmission rate R_S with high probability.



centering

Fig. 11.4. Receiver R behavior

Upon receiving an RTS, each neighbor N_i of node S must verify that the rate and additional interference of the new link are admissible. First, N_i uses the received signal strength of the RTS message to compute its current distance from S, which enables N_i to compute PL_{si}. Next, N_i calculates the received power of the new link, using the equation:

$$P_{R_i} = \frac{min(P_{allowed}, P_{max})}{PL_{si}} \tag{11.8}$$

The intended receiver R must check for two additional conditions in order before admitting the new link: (1) the received power of the new link does not exceed the MSI that R advertised in it most recent hello message; and (2) the new link has an acceptable signal to noise ratio at R. If the link request satisfies both (1) and (2), then R sends a Clear To Send (CTS) message to node S immediately, otherwise R must select the appropriate rate and/or power and include them in a Not Clear to Send (NCTS) message to S.

If the link request exceeds the declared MSI of node R, then R computes the allowable received power $P_{Rallowed}$ from neighbor S. Subsequently, node R can then compute the rate at which a signal from S arriving at R with power $P_{Rallowed}$ would have an acceptable signal quality:

$$R_r = \frac{P_{Rallowed}}{SNR_{Min} \times \mu(\eta_r + U_r)} \tag{11.9}$$

The other case is that the link request does not violate node R's MSI but fails to achieve an acceptable SNR at node R. In that case, node R uses $P_{allowed}/PL_{si}$ instead of $P_{Rallowed}$ in (11.9) to get the R_r that would result in an acceptable SNR for the new link at node R. In addition, node R could check if the TH chosen by S closely correlates with one of the TH-codes currently used by other links in its neighborhood [205].

Node S waits for incoming neighbors' replies. If S receives only a CTS message, then it sends a "Reserve" message indicating the rate, power and duration of its link reservation, and it immediately sets up a link to node R. The "Reserve" message also allows the receiver to synchronize to the sender's TH code. If at least one NCTS arrives at S, then S adjusts $P_{allowed}$ and R_S in order to satisfy the updated interference state of its neighbors. If the new value of R_S is higher than R_{Min}, then S sends a "Reserve" message and sets up the link with the newly chosen rate and power. Otherwise, the link request fails.

Upon reception of the "Reserve" message and establishment of the link, all neighbors of S update their MSI and current interference levels. If any neighbor N_i detects an appreciable variation in either of the two parameters as a result of the update, N_i issues a hello message to inform its neighbors of the state change. If any node's hello message timer expires during a link setup phase, the node postpones sending the hello message until after the link request, to avoid inconsistent views of network state during the link request.

DB traffic

The purpose of DB traffic in UWB is to support best-effort delivery of data without any quality requirements. More specifically, a DB link can sacrifice performance in order to keep interference levels constant at neighbors. In U-MAC, nodes accommodate a new DB request by lowering rates of their other active DB links so that the creation of the new link keeps the interference levels constant. Naturally, ensuring that the link rates are adaptive to network state requires the symmetric mechanism of increasing rates once a DB transmission ends. To promote fairness, the receiver can also split its DB bandwidth equally among all active DB links.

A node could monitor the traffic nature of its neighborhood through hello messages and allocate a portion of its spectrum to each traffic class. As mentioned earlier, each node allocates λ of its MSI to DB traffic, and $(1 - \lambda)$ to RB traffic. Each node further divides its DB portion equally among all active DB links, and it adjusts all DB power levels and potentially the corresponding rates whenever a new DB link is established. If a node N_i has k active DB links, a new DB link would cause it to adjust the received power level of each link based on the expression:

$$P_{Ri} = \frac{MSI_i \times \lambda_i}{k+1} \tag{11.10}$$

Since nodes in the network use random TH-codes, the aggregation of several transmissions appear as background noise at any receiver. The addition of another DB link with a new TH code does not add to the interference at a receiver if the overall DB received power stays the same.

When S has DB data to send to node R, it checks the information compiled from recent hello messages[2]. For each neighbor N_i, S uses a modified version of (11.6):

$$P_{allowed} = min\{\frac{MSI_i \times PL_{si} \times \lambda_i}{T_f \sigma^2 (k+1)}\} \tag{11.11}$$

S then proceeds as in the RB case to assign a corresponding rate with an appropriate margin, to send an RTS message, and to await neighbor replies. The intended receiver R replies with CTS if it consents to the DB request, or with NCTS if the request is not appropriate. Other neighbors of S only reply in case they do not agree with the DB request.

Once S processes all the replies, it sends a “Reserve” message and begins sending DB data. The neighbors of S that are sources to DB links hear the “Reserve” message and lower their DB link power and rates as needed to accommodate the new DB link from S. However, two-hop neighbors of S do not detect the “Reserve” message. Suppose N_j is a two-hop neighbor of S, and N_j has an active DB link with a neighbor N_i of S. When N_i detects that N_j has not reduced its power in response to the “Reserve” message, N_i signals N_j to lower the power (and potentially the rate) of its active DB transmissions.

---

[2] Since λ changes rarely, the MSI triggering of hello messages ensures that neighboring can make DB rate selections based on a sufficiently up-to-date local view of the network. To account for DB link changes at neighbors since the last hello message, S could use a margin which is dependent upon the traffic pattern.

Modifying the power of all received DB transmissions upon the creation of a new DB stream ensures that the aggregate received power from DB traffic remains constant at each node. Based on the size of DB traffic indicated in RTS and the granted rate in “Reserve”, each two-hop neighbor can set a timer to indicate the approximate time that this DB link will be active. When the timer for a DB link expires, each neighbor releases the link resources, recomputes the updated state parameters locally, and includes these changes in the next scheduled hello message. There is no need to trigger hello messages upon a DB link expiration, since all nodes in the area set the same timer for this link, and each of them releases its resources locally. Clock skews among the nodes only lead to instantaneous differences in local node states and do not affect the protocol behavior.

11.3.6 MSI Margin

So far, the discussion has focused on rate and power assignment from the point of view of a sender. Each sender must know the interference state of its neighbors when it sets up a new link. The interference state information that a node advertises in its hello messages is therefore the basis for transmission power and rate assignment at the sender. First, if N_i advertises MSI_{N_i} of its weakest active link, one neighbor N_j may set up a link with N_i that causes MSI_{N_i} to drop to zero, which would block other nodes in the vicinity from setting up new links. Thus, the first challenge is to declare an MSI value that makes efficient use of the medium and ensures fairness. The other challenge for MSI reporting is that, as we mentioned earlier, minor changes in MSI or interference at the node do not trigger hello messages, so nodes may have slightly inaccurate state information about their neighbors.

To address these challenges, each node can declare a fraction of its MSI, in order to avoid starvation of some nodes and to account for unreported minor changes. The portion of MSI that a node declares should depend on how busy a receiver it is. For example, if the average number of active links at N_i in the recent past is low, then N_i can declare a larger portion of its MSI, since it does not expect to receive many more requests. On the other hand, if N_i has many active links on average, then it advertises a smaller MSI. We use the following expression to compute the declared MSI for hello messages at node N_i:

$$MSI = \frac{MSI_{total} \times \delta}{\overline{active}} \tag{11.12}$$

where MSI_{total} is the full MSI of the weakest link at N_i, $\overline{active}$ is the number of active links at N_i, and δ is a topology dependent adjustable margin that trades off fairness for throughput. In Section 11.4, we explore the effect of varying the values of δ on fairness and throughput.

The margin δ enforces power control, which can contribute to fairness among near and far nodes. Consider the case of Fig. 11.5, where both nodes A and C wish to send data to node B. We expect $P_{allowed}$ at C to be higher

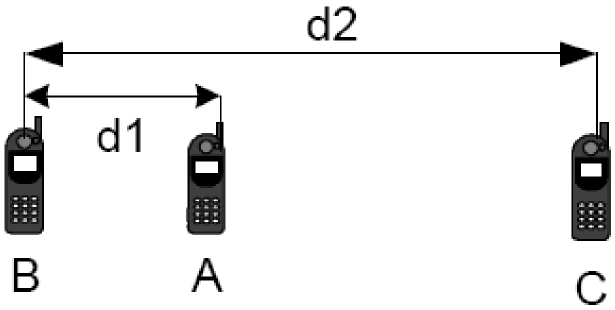


Fig. 11.5. Two senders at different distances from a receiver

than at A for an equal contribution to the interference at node B. However, the value of $P_{allowed}$ at C is constrained by two other factors: the absolute upper limit on transmit power (set by regulatory entities); and the maximum allowable interference at the neighbors of C.

The upper limit on UWB emissions affects network behavior when $P_{allowed}$ is higher than P_{max} (see (11.7)). In that case, both nodes A and C may set up a link with B at P_{max}, so the rate of a link from A to B can be up to $\frac{d_2^2}{d_1^2}$ times larger than the rate of a link from C to B.

If the interference state at node B causes the values of $P_{allowed}$ from nodes A and C to be lower than P_{max}, $P_{allowed}$ at C is constrained by the maximum allowable interference at C's neighbors. Because C is further away from B than A, C has a higher value for $P_{allowed}$ for the same MSI declared by B. It is therefore more likely that a new link between C and B at $P_{allowed}$ violates the MSI of one of C's neighbors. As a result, C selects a power level lower than $P_{allowed}$ for a link with B, which results in a lower link rate. This situation is less likely to occur at node A since $P_{allowed}$ at A is relatively low.

Thus, the relative distances of nodes in the network are a dominant factor for allocating rates in the absence of power control. In Section 11.4, we investigate the impact of distance and the MSI margin δ on throughput. To this end, nodes must choose rate and power values for new data links based on their view on past and current network state, and based on their projection of future network conditions.

11.4 Simulation and Results

We used OPNET modeler [215] to implement our protocol model, and to examine the protocol performance in two network settings with RB traffic. First, we consider a case where all the nodes communicate with one central receiver. This case is representative of personal area network settings, such as a home network in which multiple multimedia devices send high quality video or audio to a central screen or computer [216]. It is also applicable to monitoring sensor networks, which typically have a single data sink. Furthermore, this scenario illustrates the performance of our protocol for a highly loaded receiver, and analyzes the degree of favorability for nodes at varying distances from the receiver.

We also consider the case of a network with symmetric traffic that applies to typical wireless local area networks. For the symmetric traffic case, we compare the performance of U-MAC to that of the reactive approach for different traffic loads.

11.4.1 Simulation Parameters

The upper limit on R_{QoS} for our simulations is 10 Mbps, and the minimum rate of a link R_{min} is based on a uniform distribution with a maximum of 1 Mbps. The minimum acceptable SNR for any link is 14.7 dB [205]. The maximum and minimum hello periods, T_{max} and T_{min}, are 10 seconds and 1 second respectively. C_{max} and C_{min} are 2 and 0 respectively. The ratio of P_{max} at any node to the thermal noise level is 10^{20} [205]. The size in bits of hello messages, RTS, CTS, NCTS, and Reserve messages are 64, 40, 16, 32, and 88 respectively. We set the MSI threshold to 10%, and the interference threshold to 50%[3].

We assumed a free space path loss model for our simulations. The simulation results provide the upper bound of performance improvement for U-MAC in a line of sight (LOS) environment and minimal channel variation. In non-LOS cases or cases where the channel conditions vary frequently, nodes running U-MAC have to provide a higher margin for transmit power to account for potential ranging errors. Note that the SNR quality margin and the MSI margin already offset ranging errors by providing a safety margin above the minimum transmit power values. In our simulations, the SNR quality margin μ is set to 2.

The MAC layer at each node receives requests from the network layer according to a poisson process, and selects the receiver at random in the symmetric case (there is only one receiver in the loaded receiver case). If the network layer at a node S requests a new link while some other node N has a link request in progress, the new request is buffered at S until N completes its current link request. The serialization of link requests achieved by the RTS/CTS exchange ensures that the MSI and interference levels at a node remain the same during the handshaking process. Once the MAC layer fetches a link request at the head of the request queue, it attempts to send RTS and wait for replies. If RTS times out, the node sends RTS again. If there

---

[3] We set the interference threshold for triggering hello messages higher than that for MSI, because the former changes more frequently. Since changes within the threshold do not trigger hello messages, nodes use margins in their rate and power assignments (see Sect. 11.3.6).

is no reply after 3 RTS messages, the link request fails. Also, more than one node may have buffered link requests, so if all of these nodes attempt to send RTS at the same time, then collisions will occur. Thus, each node waits for a random time within 0.2 seconds before servicing a queued link request to reduce the probability of RTS collisions. A node that has just completed a link setup cycle must choose a random time within 0.3 seconds before it services any buffered requests. This mechanism helps promote fairness, since it gives the node with the most recent link a lower chance of immediately starting a new link request. Finally, each node may have multiple active links at the same time, by using a separate TH-code for each link.

11.4.2 Results

Loaded Receiver

The topology for the loaded receiver case has 25 nodes, where 24 nodes are located at distances varying from 5m to 27m from the common receiver. We observe the impact of distance from the receiver on transmission rate using our MAC protocol, and we demonstrate how power control can be used to adjust the radius of favorable senders around the receiver.

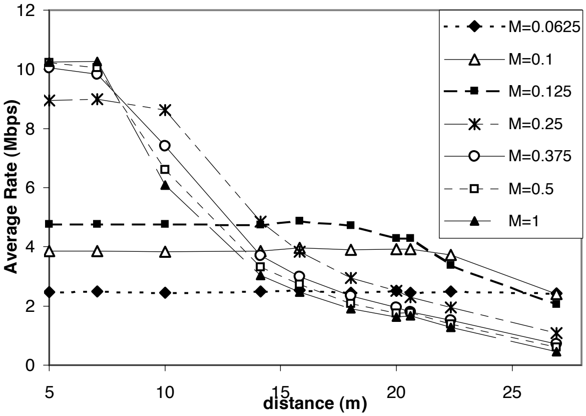


Fig. 11.6. Node throughput vs. distance

Figure 11.6 plots the average node throughput at various distances from the data sink. Let the distance of the closest node from the receiver equal d_{min}. We define the radius of fair access (ROF) as the maximum distance from the receiver within which nodes get similar throughput as nodes at d_{min} from the receiver. A node at a distance $d > ROF$ from the receiver achieves a throughput proportional to $1/(d - ROF)^2$. As we lower δ, we find that ROF expands according to the following expression:

$$\delta = \frac{1}{2^n},\ n \geq 2; \qquad (11.13)$$
$$ROF \simeq 2^{n-1} d_{min}$$

Equation 11.13 states that cutting δ by half doubles the ROF for δ values of 0.25 or lower. Figure 11.6 also shows that lower values of δ improve the performance of nodes further away from the receiver, even if these nodes remain outside the ROF. To have strictly fair access to the receiver among all nodes, ROF must equal the radius of the network centered around the common receiver.

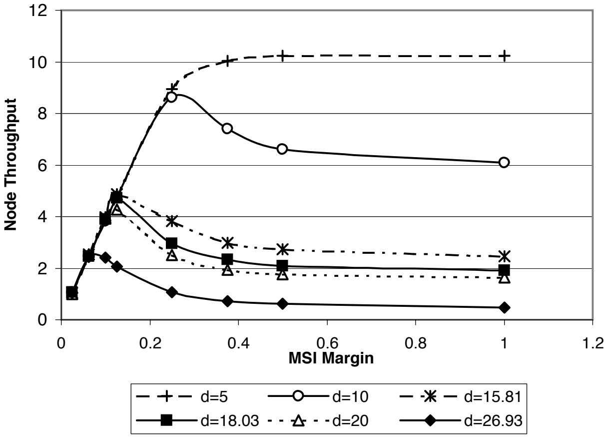


Fig. 11.7. Node throughput vs. MSI margin

Figure 11.7 plots the local throughput of nodes at different distances from the receiver against δ. The maximum value of each distance curve in Fig. 11.7 represents the highest achievable throughput for a node at that distance in this topology. For nodes that are relatively far from the receiver, the highest achievable throughput using U-MAC occurs for low values of δ. Figures 11.6 and 11.7 provide the basis for adjusting δ to favor nodes at certain ranges depending on the spatial distribution and throughput requirements of nodes in a network.

Finally, Fig. 11.8 shows how network throughput varies with δ. Decreasing δ from 1 to 0.25 improves network throughput. This peak in throughput can be understood by examining Fig. 11.6, which shows that an MSI margin δ of 0.25 widens the ROF to 10m and raises the throughput of nodes further away from the receiver with only minor decreases in the throughput of nodes closer to the receiver. Lowering δ beyond 0.25 causes significant decreases in throughput of nodes that are close to the receiver, and thus yields lower overall throughput.

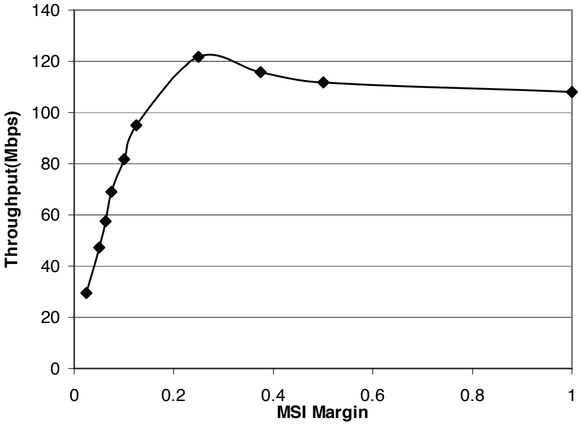


Fig. 11.8. Throughput vs. MSI margin

Symmetric Traffic

We considered three different topologies with 25 nodes each to investigate the symmetric traffic case. The first topology is a grid topology with a constant node separation distance of 5m. The second topology is a random distribution of nodes in a 100m x 100m area, with an average and minimum internode distance of 10m and 5m respectively. The third topology is a random distribution of nodes in a 50m x 50m area, where the average and minimum internode distances are 5m and 1m respectively. The results presented here are the average of three topologies.

We vary the arrival rate of new link requests to observe the behavior of the protocol for different traffic loads. The δ value used for this scenario is 1.

We first consider the link setup latency benefits of using U-MAC. In the reactive approach, a node that sends RTS must wait for replies from all of its known neighbors. Each of the neighbors uses a probabilistic back-off scheme for sending its response in order to avoid collisions of replies on the control channel. In U-MAC, a node that has sent RTS only waits for replies from the receiver and any neighbor with conflicts, so there is an inherent latency improvement. Figure 11.9 compares the link setup latency in U-MAC to the reactive approach. The average latency in U-MAC increases steadily from 13 ms at low arrival rates to 93 ms at an arrival rate of 0.66. At low link request arrival rates, the improvement in average latency of U-MAC over the reactive approach remains between 130 and 155 ms. The gap starts to widen at a request arrival rate of 0.25 and reaches a maximum of about 36 seconds at very high arrival rates. The exponential increase in latency for the reactive case is attributed to the requirement that all neighbors must send their replies upon a link request. An increased frequency of link requests causes a sharp rise in link setup latency. At arrival rates of 0.5 and higher, both the average and maximum latency for the reactive protocol stabilize. The nodes reach

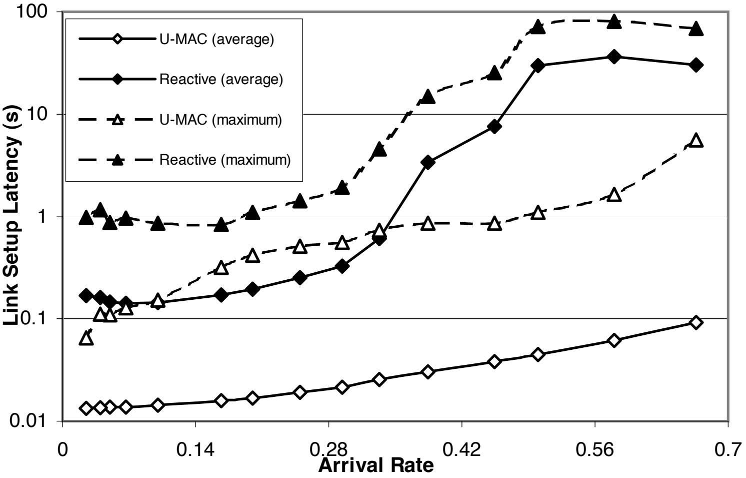


Fig. 11.9. Average link setup latency vs. offered load

their limit in the rate of requests they can handle, and although the arrival rate varies, the same number of link requests are serviced while the other link requests are discarded locally.

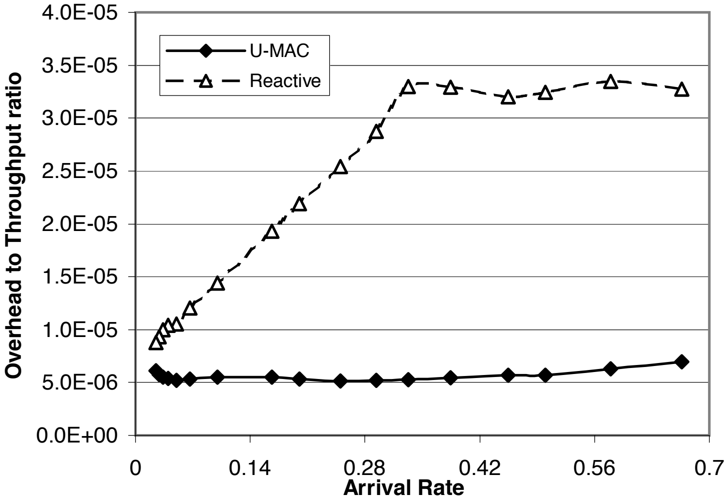


Fig. 11.10. Control overhead vs. offered load

Next we consider the control overhead of U-MAC and compare it to the reactive case. For each approach, we obtain the ratio of the bit rate used by control messages to the overall bit rate in the network, which we refer to as Overhead to Throughput ratio. Figure 11.10 reveals that this ratio for

U-MAC remains constant and only starts to increase slightly at arrival rates above 0.56. For the reactive case, the Overhead to Throughput ratio increases at a constant rate with increasing link requests because the increase in control messaging exceeds the throughput increase. At arrival rates above 0.33, the ratio in the reactive approach stabilizes as both the control overhead and the network throughput remain almost unchanged.

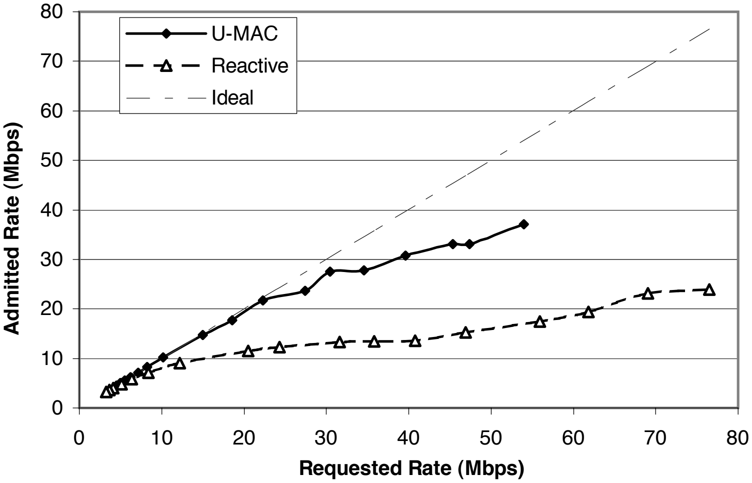


Fig. 11.11. Admitted load vs. offered load

Figure 11.11 plots the admitted rate as a function of the offered load for both U-MAC and the reactive approach. In an ideal scenario, the network would admit all of the requested transmission rate, which corresponds to the line $y = x$. In U-MAC, the admitted rate is the same as the ideal case for loads up to 20 Mbps. As the link request rate grows, state changes occur more frequently, and as a result nodes have less accurate information about their neighbors' states. Consequently, the admitted rate starts falling short of the requested rate, but the behavior remains close to the ideal case at offered loads above 20 Mbps. In the reactive approach, the requested rate is fully admitted only for loads below 10 Mbps. As the network load increases, the admitted rate in the reactive approach is increasingly lower than the requested rate. Figure 11.11 also reveals that nodes in the reactive approach request more bandwidth than in U-MAC because of their lack of information on network state. In U-MAC, nodes request only as much bandwidth as can be supported by the network according to their local view of network state.

Figure 11.12 compares the overall network throughput in both the reactive approach and U-MAC. When the link request arrival rate is low, the throughput of both cases is similar because few links are active simultaneously, so protocol mechanisms have minimal effect. As the link request arrival rate

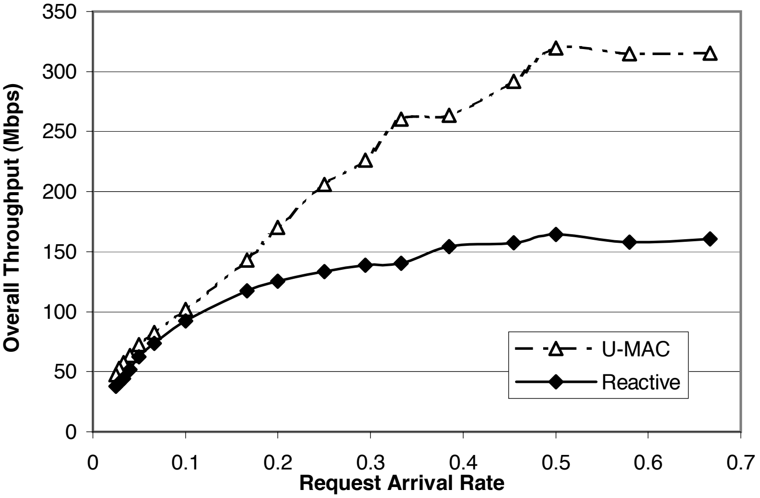


Fig. 11.12. Throughput vs. offered load

increases, nodes in the reactive case grab bandwidth greedily and limit the potential number of coexisting links. In U-MAC, nodes stay updated about the network state which allows for more efficient use of the medium. Thus, there is a growing gap in the throughput as arrival rates increase above 0.1. At a request arrival rate of 0.66, the throughput for U-MAC is about double the throughput in the reactive case.

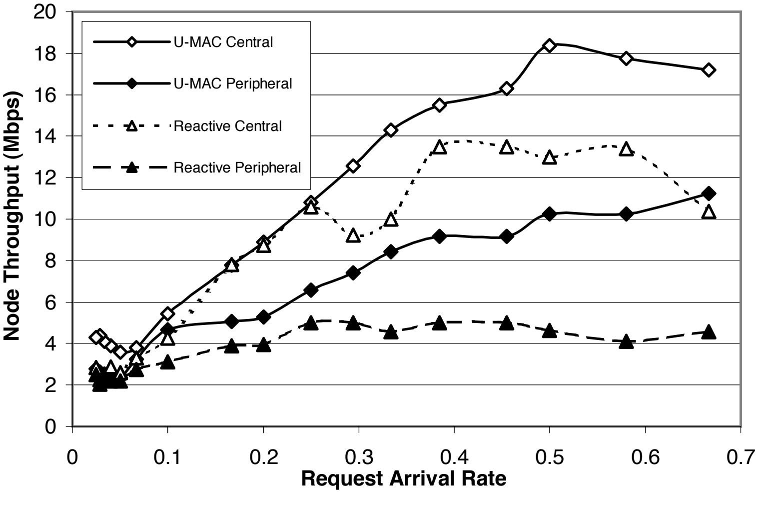


Fig. 11.13. Throughput of central and peripheral nodes

Figure 11.14 plots both the admitted link rate and the requested link rate as functions of the link request arrival rate for U-MAC and the reactive approach. In the reactive approach, the requested link rate is independent of network state, so it does not vary with network load. As a result, the gap between admitted and requested link rates grows with network load, and stabilizes for link arrival rates above 0.3. In U-MAC, nodes adapt their requested link rate to the interference and number of active links in the network for new link requests. The requested link rate is generally admitted for link request arrival rates up to 0.25. For link request arrival rates between 0.25 and 0.5, there is a growing gap between the requested and admitted link rates. This indicates that nodes make less accurate local rate and power assignments due to a higher rate of change in network state. However, the gap between the requested and admitted rate stabilizes for arrival rates between 0.5 and 0.66, which indicates that the portion of inaccurate rate and power assignments remains the same for those loads.

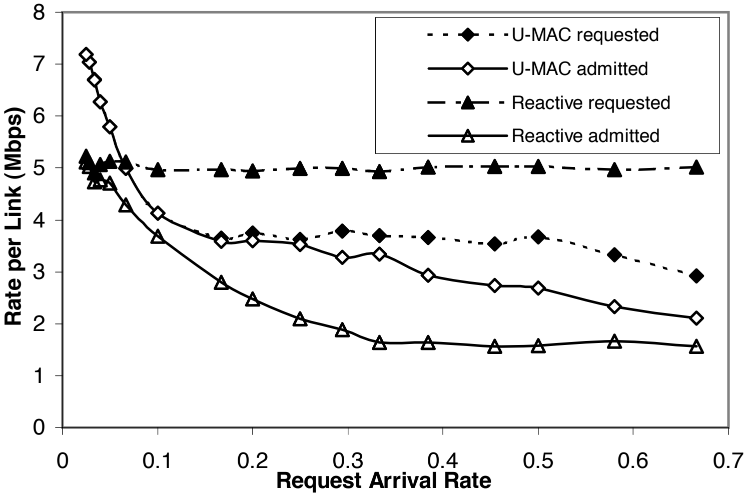


Fig. 11.14. Rate per link vs. request arrival rate

Finally, Fig. 11.13 compares the sustained throughput of a node central to the topology and a node on the periphery versus the link request arrival rate. At low arrival rates, the gap between the throughput values of the two nodes is narrow regardless of MAC protocol mechanisms. As the arrival rate increases, both nodes exhibit higher throughput, but the gap increases indicating that nodes central to the topology grab more bandwidth as the traffic load increases. The gap stabilizes for a link request arrival rate of about 0.5, where network throughput starts to saturate.

The gap in throughput between the 2 nodes is due to the difference in their average distances from other nodes in the network. Decreasing the δ

to reduce this gap is not always an attractive option in symmetric traffic networks, since it affects overall throughput more severely than in the single receiver case. Both the reactive approach and U-MAC exhibit this behavior; however, the gap for the reactive case is larger due to the greedy approach in bandwidth assignment. e

11.5 Discussion and Conclusion

Previous work has addressed the joint rate assignment problem under different assumptions and conditions. Lal and Sousa [212] proposed a reactive protocol that addresses the problem for Direct Sequence Code Division Multiple Access (DS-CDMA) networks. Their protocol involves a set of handshaking messages to negotiate resource allocation and leverages the concept of MSI along with several techniques for resource allocation based on minimizing power, maximizing rate, or maximizing SNR.

The CDMA model of Lal and Sousa was adapted by Cuomo et al. [206] for UWB networks. Their work presented a reactive approach [206] to address the joint rate power assignment problem. The approach specifies that nodes request neighbor information on-demand for setting up a new link. More specifically, a sender first polls its neighbors for their MSI measurements. Each neighbor must send an MSI measurement to the sender, and neighbor replies may overlap in time. Once the sender gets replies from all its neighbors, it selects the appropriate rate and power for the link and initiates another handshake to confirm the selected parameters. This protocol requires the sender to receive and differentiate between replies from all its neighbors at the same time. As the number of neighbors grows, so does the number of simultaneous replies that must be processed at the sender. This presents a technical challenge since all replies use the same control channel code. The work in [206] disregards this challenge and assumes all control messages are successfully received without taking up any radio resources. Ensuring that all neighbor replies are received successfully requires some probabilistic back-off scheme at each neighbor, which delays link setup. Going through two handshakes further contributes to link setup latency.

The comparison of U-MAC and the reactive approach that is similar to [206] in a realistic scenario has shown that U-MAC decreases control overhead and link setup latency considerably while making more efficient use of the medium. The decrease in control overhead and improvement in efficiency are attributed to the availability of neighbor state information locally at each node, and to the fact that only some neighbors reply to each link request. The latency decrease also benefits from the selective neighbor replies, as well as the elimination of one control message in the handshaking sequence.

The work in [207] and [208] independently proposed a proactive protocol that is related to U-MAC in that it uses periodic broadcast messages. The authors presented techniques for setting and adjusting MSI margins and simulation results for access probabilities under different medium conditions. U-MAC further expands these ideas in 2 directions. First, U-MAC dynamically sets the hello message period to adapt the degree of state changes at nodes in order to avoid unnecessary state advertisements. Second, the protocol in [207] and [208] specifies that a node setting up a link requires replies from all neighbors. In U-MAC, only the subset of the neighbors that do not agree with the link parameters send replies, which is similar to the selective reject (SREJ) concept in [212]. As a result, U-MAC reduces control overhead and link setup latency.

In addition, our work on U-MAC provides the first comparative study of reactive and proactive approaches to the joint rate and power assignment problem in UWB. The study reveals that the proactive approach doubles network throughput under high traffic conditions. Our other simulation study addresses fair access between nodes at varying distances from a common receiver, and explores the tradeoff between achieving maximum network throughput and promoting fair access to individual nodes. The results reveal that there is a radius of fair access for each receiver within which all nodes achieve comparable throughput. The MSI margin δ in U-MAC controls the radius of fair access and determines the balance between fair access to the receiver and overall network throughput.

One direction for future research is to integrate multihop links into our protocol. Using information provided by hello messages, nodes can make local decisions on least-cost paths to a particular destination [175].

Another issue for future investigation is the effect of mobility on the protocol. The protocol framework provides measures of positional reliability, but our simulations only consider stationary nodes. It would be interesting to explore techniques for using UWB's radar capability to keep track of mobile nodes, and to study the impact of mobility on the hello message period and the resulting protocol behavior.

Finally, coupling UWB's positioning capability with directional antennas [217] can reduce power consumption since the receiver captures most of the transmitted power. Once a sender knows the receiver's location, it can direct the antenna beam towards the receiver. Smart antennas [218], which have been considered for UWB transmissions [219,220], are directional antennas that can physically steer themselves towards the receiver. We can modify U-MAC to operate with smart antennas by considering interference in each sector around the receiver independently. We expect that the modified protocol would enable more simultaneous links, provided that the links are evenly distributed in all sectors.

The design philosophy of U-MAC also addresses the cross-layer design challenges from Section 7.3.3. Instead of providing a short-term solution to a specific problem, U-MAC's design adopts a forward-looking perspective for UWB networks through the extended cost function in Appendix A, which enables the incorporation of additional performance metrics into configuration decisions. The cost function provides a modular and structured tool that can be tuned by different application to customize network behavior. The MSI margin is another tunable parameter in U-MAC that controls network performance. MSI margin controls the ROF around each node, which determines the radius within which all neighbors can have fair access to the channel.

In sum, we have presented a new proactive and adaptive MAC protocol for UWB networks called U-MAC that relies on cross-layer information sharing. U-MAC provides well-defined parameters to control the fairness/throughput tradeoff while reducing control overhead and connection latency. Because the performance goals of U-MAC inherently cut across traditional layers, the parameters that control the behavior of U-MAC originate at the application, network, MAC, and physical layers. The performance evaluation in this chapter has revealed the importance of information sharing across these traditional layer boundaries for providing custom performance guarantees in UWB ad hoc networks.

12

Acoustic Underwater Sensor Network

Acoustic technology has been established as the exclusive technology that provides robust underwater communications for military and civilian applications. One particular civilian application of interest is the deployment of underwater acoustic sensor networks. The main challenges of deploying such a network are the cost and the limited battery resources of individual sensor nodes. Thus, this case study aims at maximizing the network lifetime and minimizing of power consumption of underwater sensor networks. This chapter applies Jurdak's cross-layer framework to an underwater acoustic sensor network for monitoring of a watershed in order to achieve network longevity through cross-layer interaction and internode collaboration. Figure 12.1 shows the customized framework for this case study. The focus of this chapter is the development of the optimization algorithm that maximizes battery life and minimizes energy consumption for an underwater sensor network. The optimization algorithm uses the following state parameters to estimate network lifetime: (1) internode distance; (2) transmission frequency; (3) sampling frequency; and (4) number of descendants in the routing tree. Transmission frequency and internode distances are both physical layer aspects. The sampling frequency is an application layer aspect, and the number of descendants in the routing tree is a network layer aspect. The local and neighborhood state information can be employed locally at each node to dynamically select the appropriate transmission frequency.

12.1 Introduction

Wireless acoustic communication is based on the modulation of sound waves in frequency, time, amplitude, phase, or position in order to embed data into

---

Portions reprinted, with permission, from (R. Jurdak, C. V. Lopes and P. Baldi. Battery Lifetime Estimation and Optimization for Underwater Sensor Networks. Sensor Network Operations, Wiley/IEEE Press, pp. 397–429, May, 2006.) ©2006 Wiley-IEEE

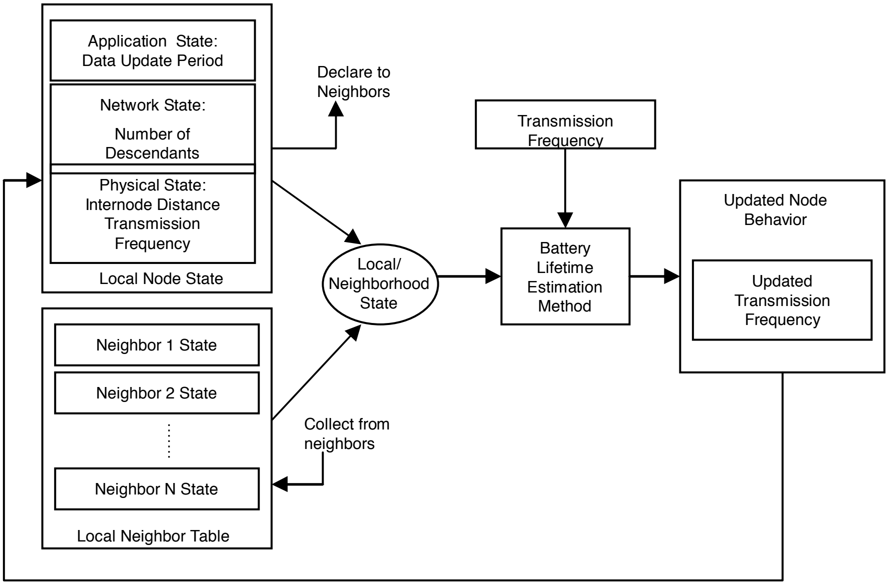


Fig. 12.1. Customized framework for underwater sensor network

the sound. Underwater acoustic communication has been used for a long time in military applications. Compared to radio waves, sound has superior propagation characteristics in water, making it the preferred technology for underwater communications. The military experience with this technology has led to increased interest for civilian applications, including the development of underwater networks. The main motivation for underwater acoustic networks is their relative ease of deployment since they eliminate the need for cables and they do not interfere with shipping activity. These networks are useful for effectively monitoring the underwater medium for military, commercial or environmental applications. Environmental applications include monitoring of physical indicators [221] (such as salinity, pressure, and temperature) and chemical/biological indicators (such as bacteria levels, contaminant levels, and dangerous chemical or biological agent levels in reservoirs and aqueducts).

The work presented in this chapter is part of an interdisciplinary effort at UCI to develop a shallow water underwater sensor network for real-time monitoring of environmental indicators, similar to current air quality monitoring systems. One of the major considerations for the development of such a network is the power consumption at individual nodes. This work is motivated by the practical need to estimate the battery life of sensor nodes, which has implications on the usefulness, topology and range of the network. Estimating the battery life of sensor networks prior to design and deployment of the actual network requires an analytical method which coarsely captures the behavior of a shallow water sensor network. On the theoretical level, this work is driven by the need to develop a generic method for battery lifetime estimation that combines both the networking and medium-specific aspects in sensor networks.

Most of the existing work has focused on modeling the battery lifetime of sensor networks in air [222, 223]. The goals of the work in this chapter are:

- To provide an estimation method for network battery lifetime specific to the conditions of underwater acoustic sensor networks
- To propose topology-dependent optimizations for power
- To show how the estimation method fits in our framework and to evaluate the benefits of the optimizations for a typical shallow water sensor network

The remainder of this chapter is as follows. Section 12.2 provides the necessary background and reviews previous related work that addresses the network lifetime issue. Section 12.3 introduces the steps of the estimation method. Section 12.4 presents the topology-specific optimizations for power consumption. Section 12.5 shows how local decisions based on topology information can estimate the global network battery life and power consumption, through two topologies that are representative of shallow water network scenarios. Section 12.6 discusses the analytical results and concludes the chapter.

12.2 Related

Interest in underwater acoustics dates back to the early 20^{th} century when sonar waves were used to detect icebergs [11]. Later, the military started using underwater acoustics for detecting submarines [11] and mines [12, 224]. Underwater acoustic applications further extended to seafloor imaging [225], object localization and tracking [226], and data communication [11] for ocean exploration and management of coastal areas. The previous experiences with underwater acoustics have led to the design of underwater sensor networks that include a large number of sensors and perform long-term monitoring of the underwater environment [227]. In underwater sensor networks, the issue of limited battery resources at the sensors is particularly important because of the difficulty and cost of recharging sensor batteries once the network is deployed.

In the recent literature, several approaches address estimation and optimization of the lifetime of energy-constrained networks. In the context of underwater networks, Fruehauf and Rice [228] propose the use of steerable directional acoustic transducers for signal transmission and reception in underwater nodes to reduce the energy consumption and thus prolong the lifetime of a node. Among other approaches that apply to more general energy-constrained networks, Tilaky et al. [229] assess the tradeoffs involved in the design and topology of sensor networks. Marsan et al. [230] consider techniques to maximize the lifetime of a Bluetooth network by optimizing network topology, and argue that their optimization techniques are also applicable to general ad hoc networks. Several routing [69,231,232,245] and MAC [56,57,179,233,234] algorithms have been developed for energy efficient behavior in sensor networks in order to maximize network lifetime. For example, Misra and Banerjee [245] present a routing algorithm to maximize network lifetime by choosing routes that pass through nodes with currently high capacity. The capacity of a node according to [245] is a combined measure of the remaining battery energy and the estimated energy spent in reliably forwarding data of other nodes in the network. Panigrahi et al. [235] derive stochastic models for battery behavior to represent realistic battery behavior in mobile embedded systems. In our work, we model battery behavior as a function of the acoustic transmit and receive power, which are the dominant sources of power consumption in underwater transceivers [236]. Some models [222,223] attempt to derive an upper bound on the lifetime of a sensor network, in terms of a generic set of parameters. Some of the parameters in our method, such as the internode distance and the number of nodes that relay data to the sink, are also considered in [222] and [223]. However, both of the previous models assume a path loss inversely proportional to d^n, where d is the distance between a sender and receiver. This assumption applies to most aerial wireless networks, but does not capture the specific conditions of underwater networks, in which the path loss depends on frequency as well as distance (see Equations 12.3-12.4 below). Furthermore, delay and multipath propagation effects in underwater networks are certainly different from aerial networks. The case of relatively infrequent data updates is addressed in [237], which focuses on radio frequency sensor networks where nodes periodically send data updates towards the central node. In our method, we also consider the case of infrequent data updates towards a central node in underwater acoustic networks, and as in [237], we attempt to derive algorithms for data gathering and aggregation that maximize the lifetime of the network.

12.3 Network Battery Life Estimation Method

The challenges of designing shallow water acoustic networks include the following:

1. Spectrum allocation: the limited available acoustic spectrum [238] in underwater environments makes this issue particularly challenging.
2. Topology: internode distances and number of forwarding nodes are factors that impact the overall performance of the network [230] [237] [222].
3. Shallow water environment: this environment tends to have distinct multipath characteristics [238] [239], for instance due to surface reflection of the signal. Shallow water noise also follows distinct patterns because of various noise sources [240], such as winds and shipping activity.

Design choices that address these challenges affect the battery lifetime of the network, which is our main metric of interest. The network battery life must be sufficiently long to avoid recharging or replacing node batteries frequently. A related metric that can be formulated is the power consumption to throughput ratio ($PCTR$), indicating the power cost of transmitting bits in the network.

Maximizing battery lifetime while minimizing $PCTR$ typically requires networks to have less frequent data updates, lower spatial density, or shorter range [229]. All of these characteristics yield lower granularity of the sensed data. Thus, there is a tradeoff involved between prolonging network lifetime and maximizing the accuracy of sensed data.

Consequently, the first step in our network battery life estimation method is to identify the design parameters that impact battery lifetime and power consumption, which are highly dependent on the network scenario. Next, the method investigates the signal propagation characteristics in the deployment region of interest as a function of the independent variables to derive the required transmission power for successful data reception. Third, we exploit the fact that data dissemination in our network is periodic and we compute the power cost of data delivery during one update period. Finally, the method uses the data delivery power cost during an update period to estimate the battery lifetime and power cost of the network. Each of the remaining subsections in this section focuses on one of these steps.

12.3.1 Network Design Parameters

Figure 12.2 illustrates our generalized network topology to analyze the tradeoffs of accurate underwater environmental indicator monitoring and power efficiency. The network in Figure 12.2 has a multihop centralized topology in which several trees are rooted at the base station, and data flow is always toward the base station. The convergence of data at the base station is appropriate for underwater sensor networks because sensor data in these networks is typically sent to shore for collection and analysis.

In the topology of Figure 12.2, nodes monitor their surrounding environmental conditions, and periodically send the collected information towards a central shore or surface station, which subsequently collects and processes the data. We consider the transmit and receive power to be the main sources of power consumption at each node [222] [237], and we assume that the sensing and processing powers are negligible.

Channel allocation is trivial for sparse networks since the data updates can be scheduled so that all nodes can use the same frequency channel at different times. However, as the network density increases, nodes must tightly synchronize their transmissions to avoid collisions on the common channel. Requiring tight synchronization among sensors adds implementation and communication cost to the network. Thus in the case of fairly dense networks, the first challenge is to provide a multiple access technique that does not rely on node synchronization and that allows simultaneous transmissions by several nodes. We consider frequency division multiplexing as a multiple access technique for

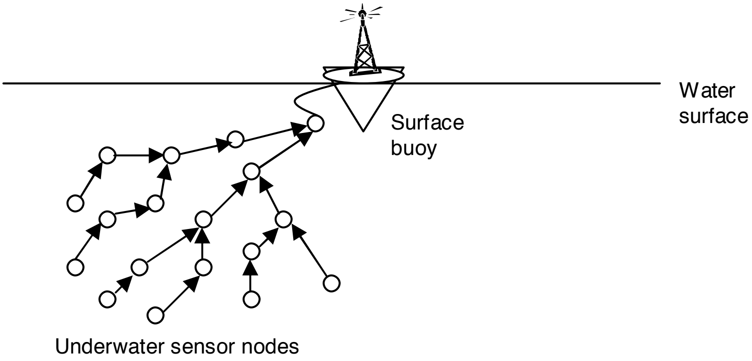


Fig. 12.2. Example underwater sensor network

our method. Because the transmissions of nodes are separated through distinct frequency channels, a node A that uses a channel with a higher frequency consumes more power than a node B using a lower frequency channel because underwater signal propagation depends on both frequency and distance (see Section 12.3.2). As a result, the battery resources at A run out earlier than the resources at B. Thus, the maximum frequency (f) is a physical layer aspect that determines the worst case for battery lifetime and power consumption of the network.

Another factor that impacts network battery lifetime and power consumption is the application layer aspect: data sampling frequency. One reasonable technique to prolong battery life is to increase the update period (R), which yields a lower power consumption rate. Significant variations in underwater medium conditions occur on the scale of a few minutes to the scale of decades [241] [242]. For example, managing a recreational beach area requires measuring danger from currents and wave sizes every several minutes. In contrast, coastal zone pollution management requires measurements in the time scale of years. Thus, an update period in the order of 20 minutes is sufficient to capture the environmental variations that occur in the shorter timescale.

To avoid consuming power for sending signals over long distances, we consider a multihop topology in which nodes that are closer to the base station[1] forward the signals of nodes further away from the base station (see Figure 12.2).

As such, nodes that are further away from the base station need only consume transmit power to get the signal to the next hop. Thus, the internode distance (d) (or the length of one hop) has significant impact on power considerations of a multihop network. A multihop topology extends the range of operation of the network, but it raises the issue of increased power overhead at intermediate nodes, which have to forward the data of nodes further away.

---

[1] A base station could be mounted on a surface buoy or on a nearby location on shore

For example, if traffic routing is based solely on distance, then the nodes closest to the base station must forward the data of all the other nodes in the network. As such, it is important that the power costs of forwarding do not overburden the forwarding nodes.

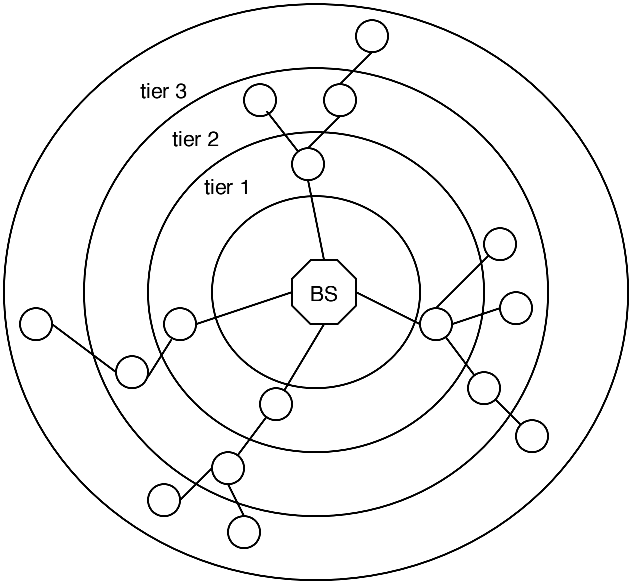


Fig. 12.3. A network with four clusters and three tiers

To address this issue, nodes are divided into clusters that are defined by proximity. Within each cluster, nodes are segmented into tiers. Figure 12.3 shows a network topology with four clusters and three tiers per cluster. The nodes at the lowest tier (tier 3 in Figure 12.3) are the furthest away from the base station and transmit messages to other nodes in the same cluster at the next higher tier (tier 2); tier 1 nodes, which are closest to the base station, finally transmit the accumulated data to the base station. Therefore, tier 1 nodes represent the bottlenecks in terms of battery lifetime, because they carry the burden of transmitting the messages of all other nodes in their respective clusters. Thus, the number of nodes in a cluster (M) is an important network layer design choice of the network. The choice of M depends on the data sampling granularity that the application requires. M also establishes a tradeoff between the power consumption for transmissions over large distances and the power overhead of forwarding data. Note that forwarding nodes could aggregate or fuse their own data [243] with data arriving from more distant nodes in order to compress the overall amount of data to be transmitted, and, ultimately, to save on transmission power. Our method does not consider aggregation, and thus presents a conservative estimate of the power consumption at the forwarding nodes.

In sum, we identified four important network design parameters that impact the battery lifetime and power consumption of an underwater sensor network: (1) the transmission frequency f; (2) the update period R; (3) the average signal transmission distance d; and (4) the number of nodes in a cluster M. Our framework can use these parameters in local algorithms to adapt each node's behavior in line with the global optimization goal of maximizing network lifetime.

12.3.2 Underwater Acoustics Fundamentals

This section covers the fundamentals of underwater acoustics, enabling the estimation of transmit and receive powers for underwater communication.

The Passive Sonar Equation

The passive sonar equation [11] characterizes the signal to noise ratio (SNR) of an emitted underwater signal at the receiver:

$$SNR = SL - TL - NL + DI \tag{12.1}$$

where SL is the source level, TL is the transmission loss, NL is the noise level, and DI is the directivity index. All the quantities in Equation 12.1 are in $dB\ re\ \mu Pa$, where the reference value of 1 μPa amounts to $0.67 \times 10^{-22}\ Watts/cm^2$ [11]. In the rest of the chapter, we use the shorthand notation of dB to signify $dB\ re\ \mu Pa$.

Factors contributing to the noise level NL in shallow water networks include waves, shipping traffic, wind level, biological noise, seaquakes and volcanic activity, and the impact of each of these factors on NL depends on the particular setting. For instance, shipping activity may dominate noise figures in bays or ports, while water currents are the primary noise source in rivers. For the purpose of this analysis, we examined several studies of shallow water noise measurements under different conditions [240] [11]. As a result, we consider an average value for the ambient noise level NL to be 70 dB as a representative shallow water case. We also consider a target SNR of 15 dB [11] at the receiver.

The directivity index DI for our network is zero because we assume omnidirectional hydrophones. Note that this is another conservative assumption, since using a directive hydrophone as described in [228] reduces power consumption.

Through the above assumptions, we can express the source level SL intensity as a function of TL only:

$$SL = TL + 85 \tag{12.2}$$

in dB.

Transmission Loss

The transmitted signal pattern has been modeled in various ways, ranging from a cylindrical pattern to a spherical one. Acoustic signals in shallow waters propagate within a cylinder bounded by the water surface and the seafloor, so cylindrical spreading applies for shallow waters. Urick [11] provides the following equation to approximate the transmission loss for cylindrically spread signals:

$$TL = 10 \log d + \alpha d \times 10^{-3} \tag{12.3}$$

where d is the distance between source and receiver in meters, α is the frequency dependent medium absorption coefficient, and TL is in dB.

Equation 12.3 indicates that the transmitted acoustic signal loses energy as it travels through the underwater medium, mainly due to distance dependent attenuation and frequency dependent medium absorbtion. Fisher and Simmons [244] conducted measurements of medium absorbtion in shallow seawater at temperatures of 4^{o}C and 20^{o}C. We derive the average of the two measurements in Equation 12.4, which expresses the average medium absorption at temperatures between 4^{o}C and 20^{o}C:

$$\alpha = \begin{cases} 0.0601 \times f^{0.8552} & 1 \leq f \leq 6 \\ 9.7888 \times f^{1.7885} \times 10^{-3} & 7 \leq f \leq 20 \\ 0.3026 \times f - 3.7933 & 20 \leq f \leq 35 \\ 0.504 \times f - 11.2 & 35 \leq f \leq 50 \end{cases} \tag{12.4}$$

where f is in Khz, and α is in dB/Km.

Through Equation 12.4, we can compute medium absorbtion for any frequency range of interest. We use this value for determining the transmission loss at various internode distances through Equation 12.3 which enables us to compute the source level in Equation 12.2 and subsequently to compute the power needed at the transmitter.

Transmission Power

We have shown how the source level SL relates to internode distance and frequency through Equations 12.2, 12.3 and 12.4. SL also relates to the transmitted signal intensity at 1 m from the source according to the following expression:

$$SL = 10 \log \frac{I_t}{1 \mu Pa} \tag{12.5}$$

where I_t is in μPa. Solving for I_t yields:

$$I_t = 10^{SL/10} \times 0.67 \times 10^{-18} \tag{12.6}$$

in $Watts/m^2$, where the constant converts μPa into $Watts/m^2$.

Finally, the transmitter power P_t needed to achieve an intensity I_t at a distance of 1 m from the source in the direction of the receiver is expressed as [11]:

$$P_t = 2\pi \times 1m \times H \times I_t \tag{12.7}$$

in Watts, where H is the water depth in m.

In short, we have presented a method to obtain the required transmitter power for signal transmissions at a given distance d and frequency f. First, we can compute the transmission loss TL in terms of f and d and we subsequently compute the source level SL, which yields the source intensity I_t. Finally, we can compute the corresponding transmit power P_t needed to achieve a source intensity of I_t.

12.3.3 Data Delivery

We now present the tier-independent method for the estimation of battery lifetime and power consumption. In Section 12.4 we consider more sophisticated tier-dependent frequency and distance assignments that build on the tier-independent method.

Without loss of generality, we assume that the size of data packets is 1 Kbit, which is enough to report 16 8-byte measurements, such as temperature, pressure, and salinity at every node in a 20 minute interval. We also assume that the bandwidth of each acoustic channel is 1 Khz. Thus, the available bit rate for each node is 1 Kbit/sec, which is well within the bit rates of current hydrophones [236], and the packet transmission time is 1 second. P_t is thus the power needed to transmit one packet in a contentionless environment. Note that a bandwidth of 1 Khz could be achieved through a combination of spread spectrum and frequency division multiplexing to achieve a higher number of coexisting nodes. Even if these multiple access techniques are used, packet collisions and corruptions remain possible. Furthermore, in each update period, a node not only sends its own data, but also the data of other nodes that are further away from a data sink.

We consider a generic Medium Access Control (MAC) protocol where a node accesses the channel, sends a data packet, and awaits an acknowledgement, which has a size of 200 bits. In the case that the acknowledgement times out, the node retransmits the data packet. Assuming a 0.1 packet loss rate, then each data packet and each acknowledgement are correctly received with a probability of 0.9. Consequently, the probability that both a packet and its corresponding acknowledgment are correctly received is 0.81, implying that each packet must be sent $1/0.81 = 1.23$ times on average. The node consumes power for sending and receiving data packets, as well as sending and receiving acknowledgements. The receive power of each message is typically around one fifth of the transmit power in commercially available hydrophones [236]. Thus, the average power in Watts consumed by a node during each update period (frame) is:

$$P_{frame} = 1.23P_t \times N(1 + \frac{1}{5} + \frac{1}{5} + \frac{1}{25}) \tag{12.8}$$

where N is the number of data packets that the node forwards during an update period. The first two terms in (12.8) account for sending and receiving data packets, while the last two terms account for sending and receiving acknowledgements.

This chapter considers two specific cases of cluster organizations: a linear chain, which represents the worst case scenario for network lifetime and applies to environmental monitoring along coastlines, rivers or aqueducts; and a grid topology, which applies to other practical environmental monitoring applications such as in a lake or bay. In the rest of this section, the discussion focuses on the chain topology, and in Section 12.5.3, we apply the method to sensors placed in a grid topology. In the chain topology, the average number of packets N forwarded by a node is equal to $M/2$ [2].

As mentioned earlier, tier 1 nodes represent the bottleneck for network battery life, since they have the highest forwarding burden of all nodes. Thus, we express the maximum amount of power consumed during one frame at a tier 1 node as:

$$P_{max} = 1.23P_t \times N_{max}(1 + \frac{1}{5} + \frac{1}{5} + \frac{1}{25}) \tag{12.9}$$

in Watts, where N_{max} is the maximum number of packets forwarded by a tier 1 node. In the chain topology, tier 1 nodes send their own data packet and forward the packets of all other nodes in the cluster during each update period, so N_{max} for this architecture is equal to M.

12.3.4 Network Lifetime and Power Consumption

A good measure of overall network power consumption is the ratio of overall power consumption to throughput. During each update period, each node in a cluster of M nodes sends its own data packet and forwards any pending data packets of its neighbors, yielding an average $PCTR$ of:

$$PCTR = \frac{M \times P_{frame}}{M \times 1000\, bits} = \frac{P_{frame}}{1000} \tag{12.10}$$

in Watts/bit. Next we want to determine the limit on the battery lifetime of a network, which depends mainly on tier 1 nodes. The time that a node's transceiver is active during one update period is important for battery life considerations. Each node uses a store and forward mechanism to forward a sequence of packets as it receives them in order to minimize the active time of its transceiver. Taking into account collisions and retransmissions, the total active time for a tier 1 transceiver in one update period is:

---

[2] This is a conservative estimate.

$$T_{total} = 1.23(N_{max} + \frac{N_{max}}{5}) \tag{12.11}$$

in seconds.

The next step is selecting a power source. We consider that we have 3 off-the-shelf 9V, 1.2 Amp-Hour batteries at each node. The total energy available at each node is:

$$E_t = 3 \times 9 \times 1.2 = 32.4 \tag{12.12}$$

in $V \cdot A \cdot hour$. The total active time of a transceiver is therefore the ratio of the total energy to the power consumed in one frame:

$$T_{active} = \frac{E_t}{P_{frame}} = \frac{32.4}{P_{frame}} \tag{12.13}$$

in hours. A node's transceiver is only active for a fraction of the time in each update period of R seconds. Therefore, the battery lifetime of a node is expressed by:

$$T_{lifetime} = \frac{T_{active}}{T_{total}} \times \frac{R}{24} \tag{12.14}$$

in days, where R is in seconds.

12.4 Topology-Dependent Optimizations

The tier-independent battery life and power consumption estimation method in Section 12.3 treats all network nodes equally, by assuming all internode distances are the same and by assigning frequency values randomly. However, the tier-independent method disregards the fact that tier 1 nodes carry a heavier power burden than other nodes. Consequently, applying measures that favor tier 1 nodes can yield improvements in battery life and power consumption. For this purpose, we propose an enhancement to the tier-independent battery life and power consumption estimation method, in which each node can locally select its own transmission frequency based on its logical position in the routing tree. This optimization is similar to the consideration of number of descendants in the routing tree for setting listening modes in the case study of Chapter 10.

Equations 12.3 and 12.4 indicate that the transmission loss increases at higher frequencies, which implies that nodes using high frequencies must transmit acoustic signals at higher power. Thus, nodes at tier 1 can choose lowest frequency band, and nodes at each subsequent tier can select the next higher frequency band. Within this rationale, nodes at the lowest tier select the highest frequency band. This assignment allows nodes with higher forwarding load to use lower frequencies and thus save power.

12.4.1 Required Modifications

One goal of tier-dependent assignments is to reduce the overall power consumption per frame in the network. Thus, tier-dependent assignments require modifications to (12.8), (12.9) and (12.11) in the general method, where N becomes:

$$N = M - i + 1 \tag{12.15}$$

for each tier i. As a result, P_{frame}, P_{max}, and T_{total} should be computed for each tier individually. We also modify the expression for $PCTR$ to reflect the distinction among tiers:

$$PCTR = \frac{\sum_{i=1}^{M} P_{frame}^{i}}{M \times 1000} \tag{12.16}$$

in Watts/bit, where P_{frame}^{i} is the power that a node at tier i consumes during one update period.

The other goal of tier-dependent assignments is to move the bottleneck tier away from the base station. Equations (12.13) and (12.14) use the individual tier values for P_{frame} and T_{total} to compute the battery lifetime of each tier. This modification shifts the dependence of the network battery lifetime from tier 1 to the bottleneck tier i.

12.5 Performance Evaluation

The requirements of our underwater environmental sensor network effort provided concrete values for some of the parameters discussed above. The deployment region of the network has a maximum depth of 10 m. To effectively monitor environmental indicators in the water, the recommended internode distances are in the range of 50 m to 1 km. The update period R is 20 minutes. Furthermore, maintenance work (such as cleaning) must be performed on the sensors themselves every 100 days or so, suggesting a target battery life of 100 days.

In the tier-independent method, we establish bounds for other parameters and analyze the results within those bounds. The maximum frequency varies from 1 Khz to 50 Khz, in steps of 1 Khz [3]. The maximum separation distance, which was established to be between 50 m and 1 km, is increased in steps of 50 m. Finally, we consider that a set of M nodes are communicating within a cluster, where M varies from 1 to 500 with a step of 1.

The rest of this section is as follows. We first derive the $PCTR$ and battery lifetime of the chain topology for each combination of distance, frequency, and cluster size using the tier-independent method. Then, we derive results for the tier-dependent assignment methods and we compare them to

---

[3] This is in line with the capabilities of existing hardware.

the tier-independent method. Finally, we estimate and compare the battery life and power consumption for a grid topology using the tier-independent and frequency-dependent methods.

12.5.1 Tier-Independent Method

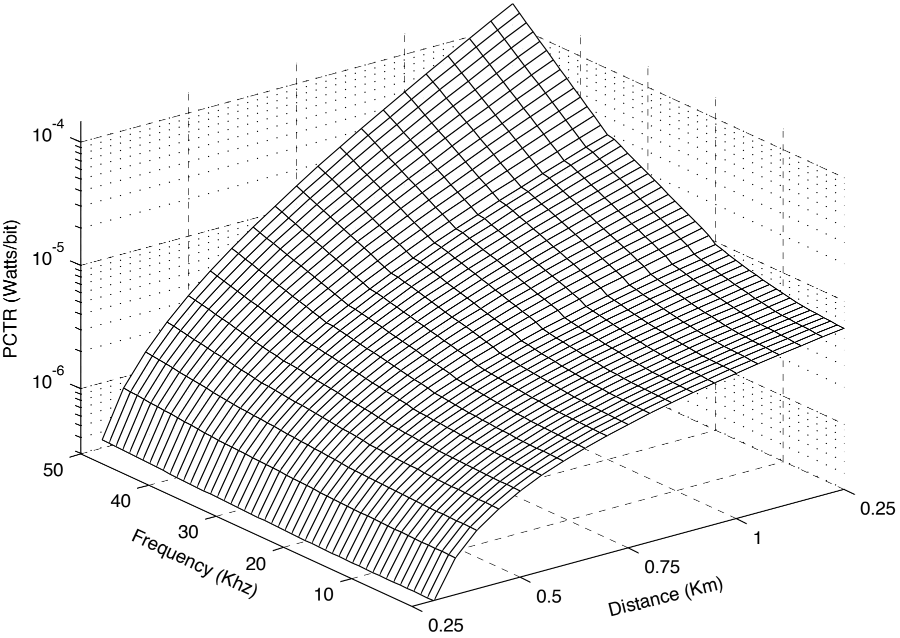


Fig. 12.4. PCTR vs. distance and frequency, for a cluster size of 500 nodes

Figure 12.4 shows the power consumption to throughput ratio ($PCTR$) plotted in terms of the maximum frequency and internode distance for a cluster size of 500 nodes. The $PCTR$ increases with higher transmission frequencies at internode distances above 250 m, whereas frequency has little effect on $PCTR$ at distances below 250 m. The maximal impact of frequency on $PCTR$ can be seen at an internode distance of 1 km, where transmission frequencies of 1 Khz and 50 Khz exhibit $PCTR$ values of 5.7 μW/bit and 148 μW/bit respectively. In contrast, varying internode distances from 50 m to 1 km does cause $PCTR$ to increase for both low and high frequencies, with the sharpest increase of $PCTR$ with distance occurring at 50 Khz.

Figure 12.5 illustrates the variation of the network battery lifetime according to the internode distance and the maximum frequency. The network

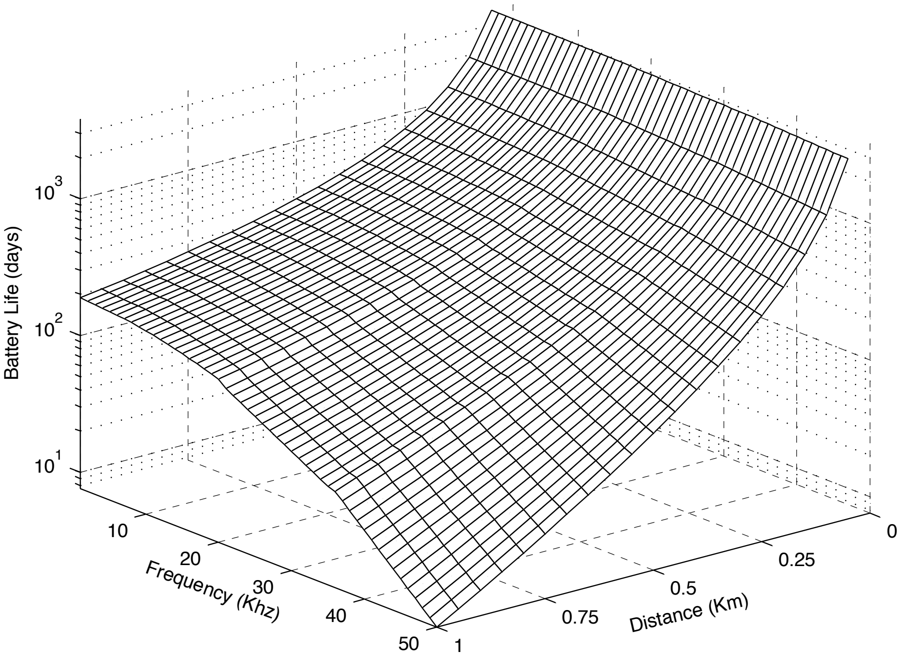


Fig. 12.5. Network battery life vs. distance and frequency for a cluster size of 500 nodes

battery life decreases sharply with increasing distance. When internode distances are small and the nodes transmit at low frequencies, the impact of medium absorption is negligible and most of the consumed power is due to signal attenuation (Equation 12.3). Medium absorbtion plays a larger role as the transmission frequency increases above 10 Khz resulting in shorter battery life. Transmitting at high frequencies over large distances shortens the battery life even further.

12.5.2 Tier-Dependent Assignments

Now we derive results for the tier-dependent assignment methods in order to compare them with the tier-independent method. Within the tier-dependent frequency assignment, we consider two subcases:

1. Constant Frequency Band (CFB): we assign tier i nodes a frequency of i Khz, as long as i is less than 50. For values of i greater than 50, all tiers use a frequency of 50 Khz.
2. Variable Frequency Bands (VFB): frequency assignments for VFB are the same as CFB for cluster sizes within 50 nodes. For cluster sizes above 50,

we divide up the spectrum into bands of $50/M$, and we assign the lowest frequency band to tier 1 nodes. Each subsequent tier uses the next higher frequency band.

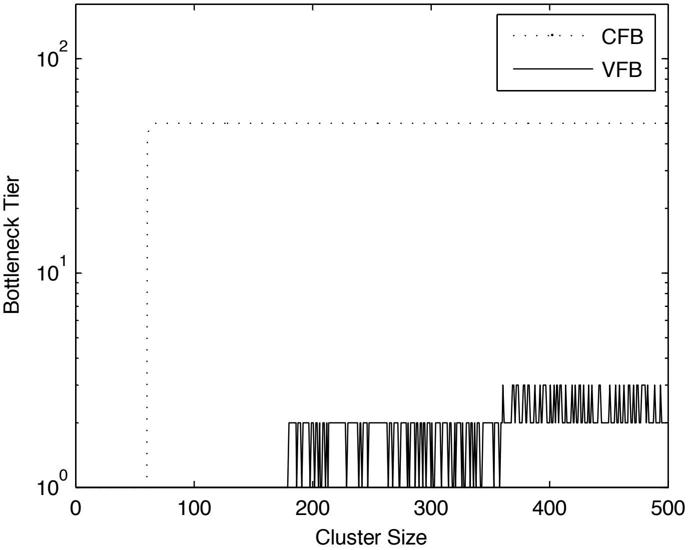


Fig. 12.6. Bottleneck tier vs. cluster size: the plots are for a distance of 1 km

Figure 12.6 provides insight into the impact of tier-dependent assignments on the tier with the shortest battery lifetime (bottleneck tier). The bottleneck tier in the Constant Frequency Band method remains at tier 1 for cluster sizes below 60 nodes. For higher cluster sizes, tier 50 becomes the bottleneck tier since nodes at tier 50 are both using the 50 Khz band (which has the highest power cost) and forwarding the data packets of other nodes. In the Variable Frequency Band method, the bottleneck tier remains at 1 for small cluster sizes, fluctuates between tiers 1 and 2 for moderate cluster sizes, and between tiers 2 and 3 for larger cluster sizes. The bottleneck tier remains close to the base station since only nodes furthest away from the base station are using the highest frequency bands.

Figure 12.7 shows the variations of the $PCTR$ for the tier-independent, CFB, and VFB cases as a function of M. The $PCTR$ in the tier-independent case increases linearly with M as a direct consequence of Equations 12.8 and 12.10. For the Constant Frequency Band case, $PCTR$ increases at a lower rate for small cluster sizes, where the maximum frequency in the network is less than 50 Khz. At cluster sizes above 50 nodes, $PCTR$ for the Constant Frequency Band case increases linearly at the same rate as the tier-independent case, since each additional tier uses the frequency of 50 Khz and thus contributes a constant portion of additional power. The two plots converge for

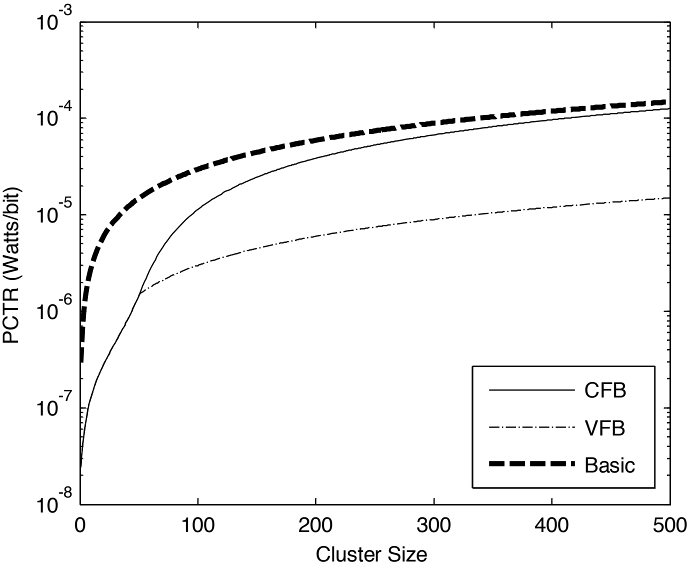


Fig. 12.7. *PCTR* vs. cluster size: The plot for the tier-independent method shows *PCTR* for a distance of 1 km and a frequency of 50 Khz. The plots for the frequency dependent assignments show *PCTR* for an internode distance of 1 km.

large cluster sizes. In the case of Variable Frequency Bands, the $PCTR$ is the same as CFB for cluster sizes below 50 nodes. However, the $PCTR$ for Variable Frequency Bands increases at a lower rate for cluster sizes larger than 50 nodes because VFB uses smaller frequency bands to accommodate additional tiers.

Figure 12.8 shows the variation of the network battery life as a function of cluster size using each of the three methods. The results in Figure 12.8 are a natural extension of the results in Figure 12.7. The CFB method yields a longer battery life than the tier-independent case for smaller cluster sizes. The improvement in battery life for VFB is more significant. For a cluster size of 500 nodes, Variable Frequency Bands yield a 24-fold improvement in network battery life.

12.5.3 Grid Topology

The estimation method uses the same equations for the grid topology as the ones for the chain topology, except for the values of N_{max} and N. In an $S \times S$ grid, N_{max} takes the value of S and N takes the value of $(S + 1)/2$.

Figure 12.9 illustrates a typical grid topology of 9 nodes. The node indices indicate the order in which nodes are placed in the grid coverage area. Once nodes form a perfect square, we begin adding sensors on tier 1 in a new column, then at tier 2, and so on, until we reach the highest tier. In Figure 12.9, once

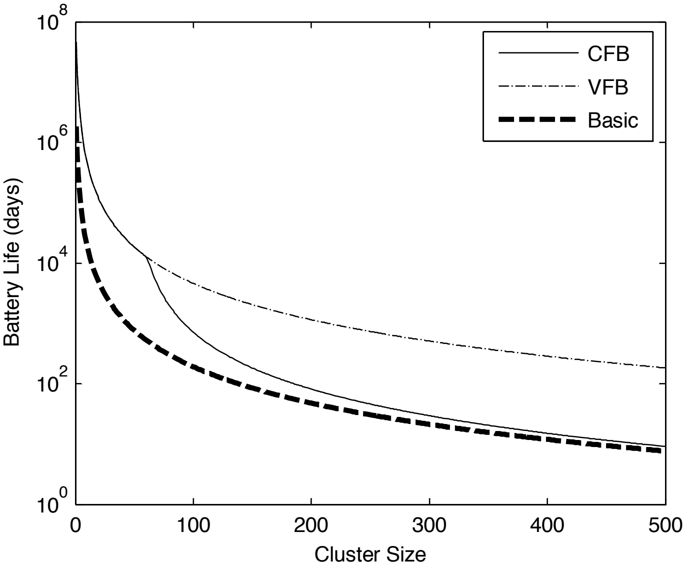


Fig. 12.8. Network Battery life vs. Cluster Size: The plot for the tier-independent method shows *PCTR* for a distance of 1 km and a frequency of 50 Khz. The plots for the frequency dependent assignments show *PCTR* for an internode distance of 1 km.

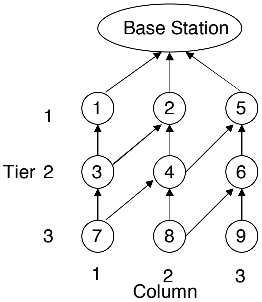


Fig. 12.9. A grid topology network with 9 nodes: The indices of nodes indicate the order in which the nodes are added to expand the network. The arrows indicate the possible forwarding paths for each node.

the first 4 nodes are in place, nodes 5 and 6 are added at tiers 1 and 2 in column 3. Once all existing tiers have a sensor in the new column, any additional sensors are placed in a new tier from left to right, until we get another perfect square topology.

Within the grid topology, nodes self-organize into a triangular lattice, as shown in Figure 12.9. This architecture allows two nodes with the same child to share the load of forwarding that child's data. Load sharing is beneficial when one of the two parent nodes has fewer children than the other, since the parent nodes can take turns in forwarding the common child's data packets. We estimate and compare the battery life and power consumption of the grid topology network for the tier-independent and the tier-dependent frequency assignment methods.

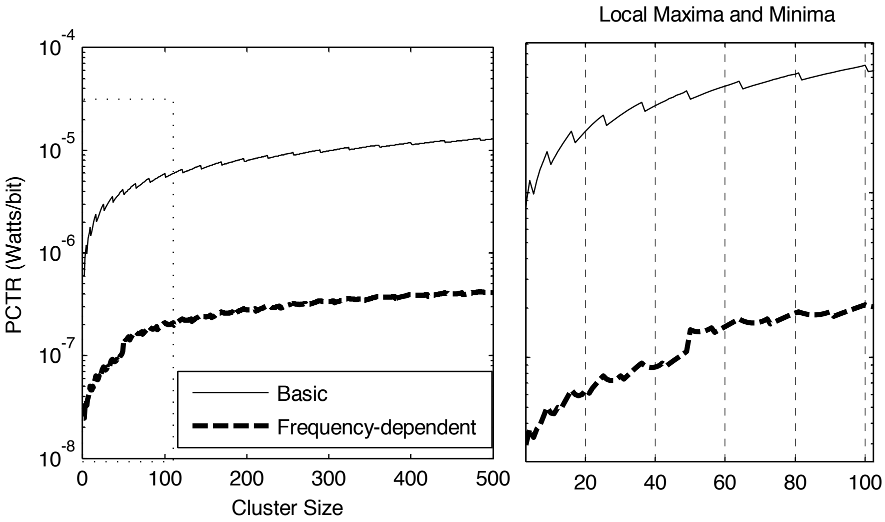


Fig. 12.10. PCTR vs. Cluster Size for the grid topology: The plot for the tier-independent method shows $PCTR$ for a distance of 1 km and a frequency of 50 Khz. The plot for the frequency-dependent assignments show $PCTR$ for an internode distance of 1 km.

Figure 12.10 shows the average power consumption in the network as the cluster size grows. An interesting observation of Figure 12.10 is the local maxima at perfect square cluster sizes. For those cases, the forwarding load is evenly split among the nodes of each tier, so load sharing does not yield any benefits. Adding an extra node to a perfect square network at tier 1 enables load sharing among the nodes of tier 1, which yields lower overall average power consumption. There are also local maxima in the plot of the frequency-dependent method at cluster sizes that correspond to a rectangular grid of size $k \times (k+1)$ for any k. To explain these local maxima, consider again Figure 12.9 for $k = 2$. There are 6 nodes in the network, with three in each tier. This symmetry among nodes of the same tier reduces the benefits of load sharing as in the perfect square case. The ratio of battery life of the tier-dependent frequency method to the tier-independent method remains constant with a 30-fold improvement for cluster sizes larger than 50. The power savings that the tier-dependent frequency method achieves over the tier-independent method grow from 0.58 μWatts/bit for small clusters to 12.5 μWatts/bit for 500 node clusters.

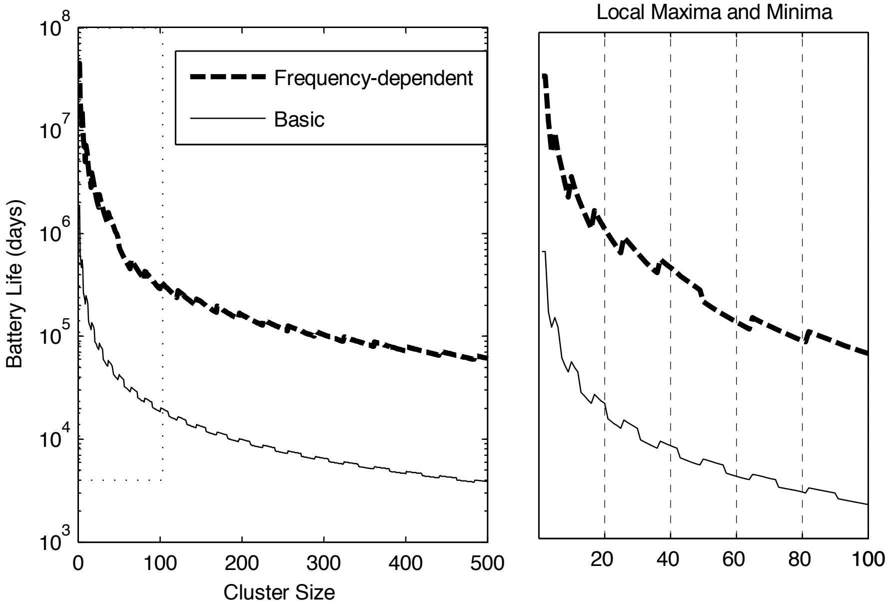


Fig. 12.11. Battery Life vs. Cluster Size for the grid topology: The plot for the tier-independent method shows *PCTR* for a distance of 1 km and a frequency of 50 Khz. The plot for the frequency-dependent assignments show *PCTR* for an internode distance of 1 km.

Figure 12.11 shows the network battery life for the tier-independent and tier-dependent frequency methods as the cluster size grows. The local minima in the plots correspond to the perfect square cluster sizes, where the power consumption peaks (Figure 12.10). In the tier-independent method, battery lifetime also drops steeply whenever adding a node corresponds to creating a new tier. In contrast, the tier-dependent frequency method does not have sharp drops for creating new tiers, primarily because tiers with high forwarding load use lower frequency bands, so the impact of nodes at a new tier is minimal. The tier-dependent frequency assignment method prolongs the battery life of the tier-independent method by a factor of 15. Even for large cluster sizes of 500 nodes in a 22×22 km^2 area, the battery life for both the tier-independent and tier-dependent methods is in the order of years, which is a significant improvement over the chain topology. This effect stems from the fact that in the grid topology, a fewer number of packets need to be forwarded by low tier nodes and neighboring nodes at the same tier can benefit from load sharing.

12.6 Discussion

The design of the battery lifetime estimation method in this chapter adopts a forward-looking design that anticipates the application of the method to other environments with different physical constraints and performance requirements. Within this long-term view, this section outlines potential scenarios for the method's application.

12.6.1 Maximum Range Alternatives

One of the requirements of our particular shallow water network is that the sensor nodes should be retrieved and cleaned every 100 days or so. This requirement implies that the network battery lifetime must be at least 100 days. We can derive the options for achieving the target battery life for the chain topology from Fig. 12.8.

Using the tier-independent method limits M to 138 nodes per cluster, which provides a network range of 138 km. The Constant Frequency Band method supports 184 nodes per cluster for a battery life of 100 days, and as a result it further extends the network range to 184 km. The Variable Frequency Band method achieves the highest network range of 500 km, with a cluster size of 500 nodes. Compared to the tier-independent method, VFB increases the cluster size, network range, and aggregated sensor data by a factor of 3.5. If we prolong the maintenance cycle to 1 year instead of 100 days, the cluster sizes of CFB, VFB and the tier-independent method drop to 120, 358, and 72 respectively.

In the grid topology, both the tier-independent and the tier-dependent frequency methods achieve a battery life of more than a year for 500 node cluster sizes, with a density of 1 node/km and a coverage area of $22 \times 22\ km^2$.

12.6.2 Method Tradeoffs

Frequency-dependent assignments are suitable for self-organizing sensor networks in which the sensors must discover the topology themselves and choose frequency bands according to their position in the topology. Constant Frequency Bands add only minimal complexity to the tier-independent scheme by requiring that nodes are aware of their position in the topology in order to choose an appropriate frequency. The Variable Frequency Band method, which achieves the longest network range, adds more signal processing complexity, since it requires the same channel rate using a smaller frequency bandwidth.

12.6.3 Grid Topology

Applying the estimation methods to a grid topology with uniformly placed nodes yielded longer network lifetime than all cases of the converging chain network, which is to be expected since the chain topology represents the lower bound on network lifetime. As mentioned earlier, networks with a grid topology are useful for environmental monitoring of lakes or bays. The estimation results that we derived cover a maximum area of $22 \times 22\ km^2$. To apply the results to larger areas, a relay station at the edge of each cluster can collect the data and forward to the base station. Alternatively, the network can still use a single base station and simply expand cluster sizes to cover the larger area.

12.6.4 Self-Recharging Sensors

Battery lifetime in sensor networks becomes less of an issue if there is some way of recharging battery resources at individual nodes without human intervention. In an underwater sensor network, nodes can derive mechanical, chemical, or solar energy from their surrounding environment. For example, nodes could absorb and store mechanical energy from water flows through small windmill-like devices. Whether the benefits of such devices overweigh the cost of building them into sensor nodes remains an open issue.

12.6.5 Method Applicability

Although we applied our method to a shallow seawater network, the method also applies to networks at any depth and any fluid. In deeper waters, the impact of both distance and frequency on transmission loss changes. One obvious distinction is that the signal undergoes spherical spreading for deeper waters, as opposed to cylindrical spreading in shallow water. Medium absorption is also depth dependent, and several studies [246] have explored this dependence through measurements. Other factors, such as the noise level, should also be modified to represent deep water environments. Applying the method to other fluids also requires similar changes to the path loss and noise models. Finally, the network deployment setting may require other changes to the method. For instance, there is no signal spreading in pipes and the transmission loss beyond a certain range is independent of distance.

Conclusion: In sum, we derived a method to estimate the battery life and power cost for underwater sensor networks. Our method first identifies the main independent variables (f, d, M, R) that impact network battery life and power consumption. Next, the method investigates the signal propagation characteristics in the deployment region of interest as a function of the independent variables (f and d in this case) to derive the required transmission power for successful data reception. Third, the transmission power estimate is combined with the relevant independent variables (M and R in this case) to compute the power cost of data delivery during one update period. Finally, the method uses the data delivery power cost during an update period to estimate the average node battery life and average network power cost.

We applied this estimation method and its tier-dependent variants to a set of shallow water network scenarios which are representative of our underwater sensor network effort. We found that for the chain topology, the Variable Frequency Band method maximizes network range for a given cluster size, provides data samples at uniform granularity, and still achieves a comparatively long battery life.

We also applied the method to a grid topology with uniformly placed sensors to estimate the network battery life and power consumption. The battery life was expectedly longer in the grid topology than the chain topology, and the tier-dependent frequency assignments prolonged battery life nearly by a factor of fifteen over the tier-independent method. Because our method is applicable to any topology or fluid medium, researchers can adapt the method to estimate power consumption and network battery life in the initial design and planning stages of fluid sensor networks.

13

Concluding Remarks and Future Directions

This book focuses on cross-layer design for ad hoc and sensor networks. The first part of the book surveys layered approaches for these networks, while the second part of the book discusses explicit cross-layer design. Part III of the book takes Jurdak's cross-layer optimization framework as an example that is applicable to three diverse case studies. The case studies cover both ad hoc and sensor networks with diverse communication technologies (RF, UWB, and acoustics) and with diverse application requirements, including energy efficiency, network longevity, throughput maximization, fair access, and low latency. The applicability of Jurdak's framework to the diverse case studies highlights the success of synergistic approaches that provide flexibility for multiple scenarios on one hand and efficient performance for each scenario on the other.

Where does cross-layer design for ad hoc and sensor networks go from here? The answer to this question points in several directions. Certainly, having an all-encompassing architecture for ad hoc and sensor networks that supports rich cross-layer interactions would promote innovation in the field and wider scale deployment of ad hoc and sensor network applications. The architectures discussed in Chapter 8 strive to provide such an architecture that provides cross-layer information sharing through a common state repository, benefiting from cross-layer interactions while maintaining the interfaces of the layered architecture. Existing cross-layer architectures still have some way to go before having a significant impact on the development of ad hoc and sensor networks. First, most proposed architectures (except for Jurdak's framework) target either ad hoc or sensor networks. Although it may limit an architecture's impact, targeting one class of networks could lead to an acceptable balance between generality and performance. Another factor that limits the impact of proposed cross-layer architectures is that most of them are still in the process of being implemented, so realizing the extent of their widespread adoption will be delayed.

A final issue is whether ad hoc and sensor networks require an all-encompassing architecture. If not, then designers can simply address each network scenario with a design coupling approach that combines the functionalities of multiple layers to improve network performance for that scenario. Many sensor networks could benefit from designing custom algorithms for each application. Because sensor networks are highly application-specific and they are typically operated by a single entity, the network user may prioritize performance for this unique application rather than the flexibility and interoperability of the algorithms for more general applications. That said, the provision of an architecture still promises to provide a fertile ground for the development and widespread deployment of more general ad hoc network applications and sensor network applications alike.

A

Extended Cost Function

Our framework determines the routing behavior of nodes by enabling each node to learn its neighborhood state and to set the routing cost of each neighbor according to the cost function. Our cost function has a weighted additive structure that enables network designers to manipulate the relevance of each cost metric by setting the metrics' weights in the cost function.

This appendix presents an extended cost function for our sensor network optimization framework, adopting many metrics from the global cost function of Baldi et al. for UWB networks. However, our cost function differs from the cost function in [175] in the following ways: (1) it assigns costs to nodes instead of links; and (2) it groups some cost metrics from [175] into composite cost metrics that are descriptive of important sensor network aspects.

Each node locally computes the cost of a neighbor N according to the global cost function:

$$\begin{aligned} C(N) &= k_3 C(delivery) + k_4 C(power) \\ &\quad + k_5 C(delay) + k_6 C(reliability) \\ &\quad + k_7 C(interference) + k_8 C(other) \end{aligned} \tag{A.1}$$

where $k_3 - k_8$ are the weights of the individual cost metrics. In what follows, we briefly discuss each of the cost function metrics.

Data Delivery

One of the main goals in almost all sensor networks is efficiently delivering sensed data to the user. In our cost function, the cost metric $C(delivery)$ indicates the data delivery cost of a particular neighbor. Delivering data in sensor networks should minimize the number of hops in the path to the data sink and maximize the quality of the links along that path. Thus, $C(delivery)$ is a composite cost metric:

$$C(delivery) = k_9 C(hops) + k_{10} C(quality) \tag{A.2}$$

where k_7 and k_8 are constants, $C(hops)$ increases linearly with the number of hops, and $C(quality)$ [175] represents the functional and communication reliability of a neighbor. In general, $C(quality)$ is inversely proportional to the quality of the link with a particular neighbor. In the absence of other performance requirements, minimizing $C(delivery)$ for each node maximizes data delivery efficiency in the network.

Power Consumption

We denote the power consumption cost at each node $C(power)$. The main causes of power consumption at a sensor node are communication, processing, and sensing. As a result, $C(power)$ is a composite cost metric:

$$\begin{aligned} C(power) = k_1 1 C(radio) + k_{12} C(processing) \\ + k_{13} C(sensing) \end{aligned} \tag{A.3}$$

where $C(radio)$ represents the power consumption of transmitting and receiving packets, listening on the channel and operating the radio in low power sleep mode. $C(processing)$ and $C(sensing)$ represent the processing and sensing power consumption respectively. These 2 metrics become relevant for routing decisions in situations where nodes assume different application roles during network operation. The constants k_{11}, k_{12}, and k_{13} depend on the hardware platform and the application.

Delay

Delay is an important cost metric for some sensor networks, such as real-time monitoring sensor networks, emergency response networks, tracking networks, or industrial automation networks. Delay in sensor networks is typically intermittent and highly variable due to sensitivity of internode communication to environmental changes. Therefore, we adopt the dynamic version of the delay cost metric $C(delay)$ in [175] which is more suitable for sensor networks.

Reliability

Reliability is crucial for applications that are sensitive to the loss of even a small percentage of data, such as an industrial automation application that monitors a factory for toxic leaks. A node's reliability measure combines the physical and communication reliability of the node [175]. In our framework for sensor networks, the $C(delivery)$ metric accounts for communication reliability. As such, we express the measure of reliability in terms of the node's physical reliability:

$$C(reliability) = \frac{k_{12}}{MTBF} \tag{A.4}$$

where $MTBF$ is the mean time between failures at a node.

Interference

Another important cost metric in some networks is interference. This metric is especially crucial in sensor networks that use spread spectrum communication technology. Our framework here aims to maintain generality with regard to communication technology, so we adopt the interference cost metric $C(interference)$ from [175] without specifying it quantitatively.

Other Metrics

Our cost function enables the inclusion of additional cost metrics, such as the cost of setting up a new link [175] or the cost of maintaining fairness. In addition, networks with multiple traffic classes may give priority to high rate nodes, which necessitates adapting the routing cost accordingly. The above metrics as well as any other metrics can be included in the metric $C(other)$.

References

1. ISO. “Open Systems Interconnection–Basic Reference Model” ISO 7498.
2. T. Socolofsky and C. Kale. A TCP/IP Tutorial. Request for Comments: 1180, 1991. available: http://rfc.sunsite.dk/rfc/rfc1180.html.a
3. Global System for Mobile Communications. ETSI standard. available: http://www.etsi.org.
4. National Archives and Records Administration. U.S. GLOBAL POSITIONING SYSTEM POLICY. March 29, 1996.
5. IEEE 802.11 WG. “Wireless Lan Medium Access Control (MAC) and Physical Layer (PHY) Specifications”, Standard. 1999.
6. “*The Bluetooth Special Interst Group*,” 1999.
7. Mica Mote Series. Crossbow Technology Inc. available: http://www.xbow.com/Products/productdetails.aspx?sid=72.
8. Zigbee Alliance. available: www.zigbee.org.
9. RFID Journal. available: www.rfidjournal.com.
10. M. Z. Win and R. Scholtz. “Impulse Radio: How it Works”. *IEEE Communications Letters*, 2(1), 1998.
11. R. J. Urick. *Principles of Underwater Sound.* McGraw-Hill, 1983.
12. J. Groen, J.C. Sabel and A. Htet. “Synthetic aperture processing techniques applied to rail experiments with a mine hunting sonar”. In Proc. *UDT Europe*, 2001.
13. C. V. Lopes and P. Aguiar. “Acoustic Modems for Ubiquitous Computing,” *IEEE Pervasive Computing, Mobile and Ubiquitous Systems,* Summer 2003.
14. R. Jurdak, C. V. Lopes and P. Baldi. Battery Lifetime Estimation and Optimization for Underwater Sensor Networks. *Sensor Network Operations*, Wiley/IEEE Press, pp. 397–429, May, 2006.
15. R. Jurdak, C.V. Lopes, and P. Baldi. “An Acoustic Identification Scheme of Location Systems,” In Proc. *ICPS’04*, Beirut, Lebanon, 2004.
16. A. Mandal, C. V. Lopes, T. Givargis, A. Haghighat, R. Jurdak, and P. Baldi. “Beep: 3D Indoor Positioning Using Audible Sound,” Proceedings of the *IEEE Consumer Communications and Networking Conference (CCNC05).* Las Vegas, Nevada, 2005.
17. A. J. Viterbi. *CDMA : Principles of Spread Spectrum Communication.* Prentice Hall, 1995.
18. J. Schiller. *Mobile Communications*, Addison-Wesley, 2000.

19. A. Chandra, V. Gummalla and J. O. Limb. "Wireless Medium Access Control Protocols." *IEEE Communications Surveys [Online].* available: http://www.comsoc.org/pubs/surveys. Second Quarter 2000.
20. M. J. Crocker. *Handbook of Acoustics*, Wiley-Interscience, 1998.
21. P. Baldi, L. De Nardis, and M. G. Di Benedetto. "Modelling and Optimization of UWB Communication Networks Through a Flexible Cost Function," *IEEE Journal on Selected Areas in Communications*, 20(9):1733–1744, 2002.
22. C. V. Lopes and P. Aguiar. "Acoustic Modems for Ubiquitous Computing," *IEEE Pervasive Computing, Mobile and Ubiquitous Systems,* Summer 2003.
23. M. Weiser. "The computer for the 21^{st} century," *Scientific American*, 265(3:94–104, 1991.
24. L. Kleinrock and F. A. Tobagi. "Packet Switching in Radio Channels: Part I-Carrier Sense Multiple-Access Models and their Throughput-Delay Characteristics," *IEEE Transactions on Communications*, 23(12):1400–1416, 1975.
25. F. A. Tobagi and L. Kleinrock. "Packet Switching in Radio Channels: Part II-The Hidden Terminal Problem in Carrier Sense Multiple-Access and the Busy-Tone Solution," *IEEE Transactions on Communications*, 23(12):1417–1433, 1975.
26. "IEEE/ANSI Standard:Carrier Sense Multiple Access with Collision Detection," 1985.
27. D. J. Goodman, R. A. Valenzuela, K. T. Gayliard, and B. Ramamurthi, "Packet Reservation Multiple Access for Local Wireless Communications," In *Vehicular Technology Conference*, pp. 701–706, IEEE, 1988.
28. P. Karn. "MACA - A New Channel Access Protocol for Packet Radio," In *ARRL/CRRL Amateur Radio Ninth Computer Networking Conference*, pp. 134–140, 1990.
29. V. Bharghavan, A. Demers, S. Shenker, and L. Zhang. "MACAW: A Media Access Protocol for Wireless LAN's," In *ACM SIGCOMM94*, pp. 212–225, ACM, 1994.
30. C. L. Fullmer and J. J. Garcia-Luna-Aceves. "Floor Acquisition Multiple Access (FAMA) for Packet-Radio Networks," In *Conference on Applications, Technologies, Architectures and Protocols for Computer Communication (SIGCOMM)*, pp. 262–273, 1995.
31. F. Talucci, M. Gerla, and L. Fratta. "MACA-BI (MACA By Invitation) A Receiver Oriented Access Protocol for Wireless Multihop Networks," In *'Waves of the Year 2000' PIMRC '97 The 8th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, volume 2, pp. 435–439, 1997.
32. C. Zu and M. S. Corson. "A Five-Phase Reservation Protocol (FPRP) for Mobile Ad Hoc Networks," In *Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 1, pp. 322–331, 1998.
33. S. Singh and C.S. Raghavendra. "Power Efficient MAC Protocol for Multihop Radio Networks," In *The Ninth IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, volume 1, 1998.
34. M. J. Markowski and A.S. Sethi. "Fully Distributed Wireless MAC Transmission of Real-Time Data," In *Fourth IEEE Real-Time Technology and Applications Symposium*, pp. 49–57, 1998.
35. Z. Tang and J. J. Garcia-Luna-Aceves, "Hop Reservation Multiple Access (HRMA) for Multichannel Packet Radio Networks," In *International Conference on Computer Communications and Networks*, pp. 388–395, 1998.

36. A. Nasipuri, J. Zhuang, and S. R. Das. “A Multichannel CSMA MAC Protocol for Multihop Wireless Networks,” In *Wireless Communications and Networking Conference*, pp. 1402–1406. IEEE, 1999.
37. L. Bononi, M. Conti, and L. Donatiello. “Distributed Contention Control Mechanism for Power Saving in Random-Access Ad-Hoc Wireless Local Area Networks,” In *International Workshop on Mobile Multimedia Communications*, pp. 114–123, IEEE, 1999.
38. J. J. Garcia-Luna-Aceves and A. Tzamaloukas. “Reversing the Collision-Avoidance Handshake in Wireless Networks,” In *ACM/IEEE Mobicom ’99*, 1999.
39. I. Chlamtac, et al. “ADAPT: A Dynamically Self-Adjusting Media Access Control Protocol for Ad Hoc Networks,” In *IEEE Globecom*, 1999.
40. Z. Tang and J. J. Garcia-Luna-Aceves. “A Protocol for Topology-Dependent Transmission Scheduling in Wireless Networks,” In *IEEE WCNC*, 1999.
41. K. T. Jin and D. H. Cho. “A MAC Algorithm for Energy-Limited Ad-Hoc Networks,” In *52nd Vehicular Technology Conference*, volume 1, pp. 219–222, IEEE, 2000.
42. C. K. Toh, V. Vassiliou, G. Guichal, and C. H. Shih. “MARCH: A Medium Access Control Protocol for Multihop Wireless Ad Hoc Networks,” In *MILCOM 21st Century Military Communications Conference*, volume 1, pp. 512–516, 2000.
43. A. Tzamaloukas and J. J. Garcia-Luna-Aceves. “A Channel-Hopping Protocol for Ad-Hoc Networks,” In *Ninth International Conference on Computer Communications and Networks*, pp. 142–147, 2000.
44. C. W. Ahn, C. G. Kang, and Y. Z. Cho. “Soft Reservation Multiple Access with Priority Assignment (SRMA/PA): A Novel MAC Protocol for QoS-Guaranteed Integrated Services in Mobile Ad-hoc Networks,” In *52nd Vehicular Technology Conference*, volume 2, pp. 942–947, IEEE, 2000.
45. Y. C. Tseng, S. L. Wu, C. Y. Lin, and J. P. Sheu, “A Multi-Channel MAC Protocol with Power Control for Multi-Hop Mobile Ad Hoc Networks,” In *International Conference on Distributed Computing Systems Workshop*, pp. 419–424, 2001.
46. J. Monks, J. P. Ebert, A. Wolisz, and W. M. Hwu, “A Study of the Energy Saving and Capacity Improvement Potential of Power Control in Multi-Hop Wireless Networks,” In *26th Annual IEEE Conference on Local Computer Networks*, pp. 550–559, 2001.
47. H. Hassanein and A. Safwat. “Virtual Base Stations For Wireless Mobile Ad Hoc Communications: An Infrastructure for the Infrastructure-less,” *International Journal of Communication Systems*, (14):763–782, 2001.
48. J. Kim and N. Bambos. “Power-Efficient MAC Scheme using Channel Probing in Multi-rate Wireless Ad Hoc Networks,” In *IEEE Vehicular Technology Conference*, volume 4, pp. 2380–2384, 2002.
49. D. Lal, R. Toshniwal, R. Radhakrishnan, D. P. Agrawal, and J. Caffery. “A Novel MAC Layer Protocol for Space Division Multiple Access in Wireless Ad Hoc Networks,” In *Eleventh International Conference on Computer Communications and Networks*, pp. 614–619, 2002.
50. Y. C. Tseng, C. M. Chao, S. L. Wu, and J. P. Sheu. “Dynamic Channel Allocation with Location Awareness for Multi-Hop Mobile Ad Hoc Networks,” *Computer Communications*, 25:676–688, 2002.

51. K. T. Jin and D. H. Cho. “Multi-Code MAC for Multi-Hop Wireless Ad hoc Networks,” In *Vehicular Technology Conference*, volume 2, pp. 1100–1104, IEEE, 2002.
52. M. Chatterjee, S. K. Das, and D. Turgut. “WCA: A Weighted Clustering Algorithm for Mobile Ad Hoc Networks,” *Cluster Computing*, (5):193–204, 2002.
53. Z. J. Haas and J. Deng. “Dual Busy Tone Multiple Access (DBTMA)- A Multiple Access Control Scheme for Ad Hoc Networks,” *IEEE Transactions on Communications*, 50(6):975–985, 2002.
54. R. R. Choudhury, X. Yang, R. Ramanathan, and N. H. Vaidya. “Using Directional Antennas for Medium Access Control in Ad Hoc Networks,” In *Mobicom*, 2002.
55. S. Jiang, J. Rao, D. He, X. Ling, and C. C. Ko. “A Simple Distributed PRMA for MANETs,” *IEEE Transactions on Vehicular Technology*, 51:293–305, 2002.
56. J. Polastre, J. Hill, and D. Culler. “Versatile Low Power Media Access for Wireless Sensor Networks”. Proceedings of *ACM SenSys*, 2004.
57. W. Ye, J. Heidemann and D. Estrin. “An Energy-Efficient MAC Protocol for Wireless Sensor Networks”. In Proc. *IEEE Infocom*, 2002.
58. T. Van Dam and K. Langendoen. “An adaptive energy-efficient MAC protocol for wireless sensor networks”. In Proc. *1st international conference on Embedded networked sensor systems*, 171-180, 2003.
59. Royer, E.M. and Chai-Keong, T. “A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks”. *IEEE Personal Communications*, 6(2):46–55, 1999.
60. L. Qin and T. Kunz. “Survey on Mobile Ad Hoc Network Routing Protocols and Cross-Layer Design”. Carleton University, Systems and Computer Engineering, Technical Report SCE-04-14, August 2004.
61. Q. Jiang and D. Manivannan. “Routing Protocols for Sensor Networks”. In Proc. *IEEE (CCNC)*, 2004.
62. G. Malkin. *RIP Version 2 Carrying Additional Information.* Request for Comments: 1723, 1994. available: http://www.ietf.org/rfc/rfc1723.txt.
63. M. A. Sportack. *IP Routing Fundamentals.* Indianapolis: Cisco Press, 1999.
64. F. Baker and R. Coltun. OSPF Version 2 Management Information Base. Request for Comments: 1850, 1995. available: http://www.ietf.org/rfc/rfc1850.txt.
65. Charles Perkins. “Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers”. In Proc. *ACM SIGCOMM’94 Conference on Communications Architectures, Protocols and Applications*, 1994.
66. T. Clausen and P. Jacquet. “Optimized Link State Routing Protocol (OLSR)”. Request for Comments: 3626, 2003. available: http://www.ietf.org/rfc/rfc3626.txt.
67. W. Heinzelmann, J. Kulik, and H. Balakrishnan. “Negotiation-based Protocols for Disseminating Information in Wireless Sensor Networks”. In Proc. *Mobicom*, 1999.
68. W. Heinzelmann, A. Chandrakasan, and H. Balakrishnan. “Energy-Efficient Communication Protocol for Wireless Microsensor Networks”. In Proc. *HICSS*, 2000.
69. S. Lindsey and C. Raghavendra. “PEGASIS: Power-Efficient Gathering in Sensor Information Systems”. In Proc. *ICC*, 2001.
70. A. Manjeshwar and D. Agrawal. “TEEN: A Routing Protocol for Enhanced Efficiency in Wireless Sensor Networks”. In Proc. *IPDPS*, 2001.

71. C. Perkins, E. Belding-Royer, and S. Das. "Ad hoc On-Demand Distance Vector (AODV) Routing". Request for Comments: 3561, 2003. available: http://www.ietf.org/rfc/rfc3561.txt.
72. Charles E. Perkins. "DSR: The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks". In *Ad Hoc Networking*, edited by Charles E. Perkins, Chapter 5, pp. 139-172, Addison-Wesley, 2001.
73. K. Sundaresan and R. Sivakumar. "Routing in Ad-Hoc Networks with MIMI Links". In Proc. *13th IEEE International Conference on Network Protocols (ICNP'05)*, November, 2005.
74. IEEE 802.11n. IEEE Standard for Enhancements for Higher Throughput. Draft Standard.
75. IEEE 802.16. IEEE Standard for Local and metropolitan area networks - Part 16: Air Interface for Fixed Broadband Wireless Access Systems. Standard. available: http://standards.ieee.org/getieee802/download/802.16-2004.pdf.
76. V. D. Park and M. S. Corson. Temporally-ordered routing algorithm (tora). Internet draft RFC, version 1, 1997.
77. M. R. Pearlman and Z. J. Haas. "Determining the optimal configuration for the zone routing protocol" *IEEE J. Select. Areas Commun.*, vol. 6, no. 2, pp. 1395-1414, Aug. 1999.
78. C. Intanagonwiwat, et al. "Directed Diffusion for Wireless Sensor Networking". *IEEE/ACM Trans. on Networking*, 11:2-16, 2003.
79. D. Braginsky and D. Estrin. "Rumor Routing Algorithm for Sensor Networks". In Proc. *1st ACM Int. Workshop on Wireless Sensor Nets and Applications*, 2002.
80. F. Ye, et al. "A Scalable Solution to Minimum Cost Forwarding in Large Sensor Networks". In Proc. ICCCN, 2001.
81. P. Chowdavarapu and V.-G. Puram. "An Overview of Routing Protocols in AdHoc Networks". Technical Report CS-90 Voice Over IP Lab, University of Kentucky Laboratory for Advanced Networking, December 2000. available: http://voip.netlab.uky.edu/ṽenu/cs690/adhocprots.doc.
82. M. Jiang, J. Li, Y. C. Tay. Cluster Based Routing Protocol (CBRP). Functional Specification Internet Draft, draft-ietf-manet-cbrp.txt, work in progress, June 1999.
83. G. Pei, M. Gerla, X. Hong and C. C. Chiang. "A Wireless Hierarchical Routing Protocol with Group Mobility". In Proc. *IEEE Wireless Communications and Networking Conference*, 1999.
84. G. Montenegro. Reverse Tunneling for Mobile IP, revised. Request for Comments: 2344, 1995. available: http://www.ietf.org/rfc/rfc2344.txt.
85. P. Bahl and V. Padmanabhan. "RADAR: An In-Building RF-Based User Location and Tracking System," In *Proc. IEEE Infocom 2000,* IEEE CS Press, pp. 775-784, 2000.
86. N.B. Priyantha, A. Chakraborty, and H. Balakrishnan. "The Cricket Location-Support System," *Proc. 6th Ann. Intl Conf. Mobile Computing and Networking (Mobicom 00),* ACM Press, pp. 32-43, 2000.
87. B. N. Karp, and H. T. Kung "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks," In proc. *ACM/IEEE International Conference on Mobile computing and networking (MobiCom '00)*, pp. 243-254, August 2000.
88. Y.B. Ko and N. Vaidya. "Location-aided Routing (LAR) in mobile ad hoc networks". In Proc. *ACM/IEEE International Conference on Mobile Computing and Networking*, pp. 66-75, 1998.

89. J. Li, J. J. Douglas, S. J. De Coutu, D. R. Karger, R. Morris. “A Scalable Location Service for Geographic Ad Hoc Routing,” In Proc. *ACM/IEEE International Conference on Mobile Computing and Networking*, pp. 120–130, 2000.
90. K. Seada, M. Zuniga, A. Helmy, and B. Krishnamachari. “Energy-Efficient Forwarding Strategies for Geographic Routing in Lossy Wireless Sensor Networks”. In Proc. SenSys’04, November, 2004.
91. M. Zuniga and B. Krishnamachari. “Analyzing the Transitional Region oin Low Power Wireless Links”. In Proc IEEE SECON’04, 2004.
92. C. Perkins. *Ad Hoc Networking*, Addison-Wesley, December, 2000.
93. E. M. Belding-Royer. “Multi-level Hierarchies for Scalable Ad hoc Routing.” *Wireless Networking (WINET)*, 9(5), pp. 461-478, Sept. 2003.
94. Dashy, S., Bandar, S., Brown, T.X. “An on-demand minimum energy routing protocol for a wireless ad hoc network”. *ACM Mobile Computing and Communication*, 6(3):50-66, 2002.
95. Rodoplu, V., Meng, T.H. “Minimum Energy Mobile Wireless Networks”. In Proc. *IEEE International Conference on Communications*, 3:1633-1693, 1998.
96. Singh, S., Woo, M., Raghavendra, C. S. “Power-aware Routing in Mobile Ad Hoc Networks”. In proc. *MobiCom98*, ACM/IEEE, pp.181–190, 1998.
97. Toh, C.K. “Maximum battery life routing to support ubiquitous mobile computing in wireless ad hoc networks”. *IEEE Communications Magazine*, 39:138-147, 2001.
98. R. C. Shah and J. M. Rabaey. “Energy Aware Routing for Low Energy Ad Hoc Sensor Networks”. In Proc. *IEEE Wireless Communications and Networking Conference (WCNC)*, March 2002.
99. D. S. J. De Couto, D. Aguayo, J. Bicket, and R. Morris. “A High-Throughput Path Metric for Multi-Hop Wireless Routing” In Proc.*Mobicom’03*, September 2003.
100. R. Jurdak. “Modeling and Optimization of Ad Hoc and Sensor Networks,” Bren School of Information and Computer Science, University of California Irvine. Ph.D. Dissertation. September, 2005.
101. C. Park, K. Lahiri, and A. Raghunathan. “Battery Discharge Characteristics of Wireless Sensor Nodes: An Experimental Analysis”. In Proc. *IEEE SECON*, 2005.
102. R. Sivakumar, P. Sinha and V. Bharghavan. “Core extraction distributed ad hoc routing (CEDAR) specification,” IETF Internet draft draft-ietf-manet-cedar-spec-00.txt, 1998.
103. K. Sohrabi, J. Gao, V. Ailawadhi, and G. J. Potie. “Protocols for self-organization of a wireless sensor network,” *IEEE Personal Communications*, pp. 16-27, October 2000.
104. K. Akkaya and M. Younis. “An Energy-Aware QoS Routing Protocol for Wireless Sensor Networks,” In Proc. *23rd International Conference on Distributed Computing Systems Workshops (ICDCSW’03)*, pp. 710, 2003.
105. S. Chen and K. NahrStedt. “Distributed Quality of Service Routing in Ad Hoc Networks,” *IEEE Journal on Selected Areas in Communications*, 17(8),pp. 1488–1505, Aug. 1999.
106. D. Ganesan, R. Govindan, S. Shenker, and D. Estrin. “Highly Resilient Energy-Efficient Multipath Routing in Wireless Sensor Networks,” *ACM Mobile Computing and Communicaitons Review*, 5(4), Oct. 2001.

107. A. Nasipuri and S. R. Das. “On-Demand Multipath Routing for Mobile Ad Hoc Networks”. In Proc. *Intl. Conf. on Computer Comm. and Networks (IC3N)*, pp. 64–70, Oct. 1999.
108. S. De, C. Qiao, and H. Wu. “Meshed Multipath Routing: An Efficient Strategy in Sensor Networks”. In Proc. *Wireless Communications and Networking (WCNC)*, vol. 3, March 2003.
109. Transmission Control Protocol. Request for Comments: 793, 1981. available: http://www.faqs.org/rfcs/rfc793.html.
110. J. Postel. User Datagram Protocol. Request for Comments: 768, 1980. available: http://www.faqs.org/rfcs/rfc768.html.
111. J. C. Hoe. “Improving the Start-Up Behavior of a Congestion Control Scheme for TCP,” In Proc. *ACM SIGCOMM 1996,* pp. 270-280, August 1996.
112. M. Mathis, J. Mahdavi, S. Floiyd, and A. Romanow. “TCP Selective Acknowledgment Options,” *IETF RFC 2018*, October 1996.
113. S. Floyd, J. Mahdavi, M. Mathis, and M. Podolsky. “An Extension to the Selective Acknowledgment (SACK) Option for TCP,” *IETF RFC 2883*, July 2000.
114. G. Holland and N. Vaidya. “Analysis of TCP Performance over Mobile Ad Hoc Networks,” In Proc. *ACM MOBICOM 1999,* pp. 219-230, August 1999.
115. K. Chandran, S. Raghunathan, S. Venkatesan, and R. Prakash. “A Feedback-Based Scheme for Improving TCP Performance in Ad Hoc Wireless Networks,” *IEEE Personal Communications Magazine,* vol. 8, no. 1, pp. 34–39, February 2001.
116. D. Kim, C. K. Toh, and Y. Choi. “TCP-BuS: Improving TCP Performance in Wireless Ad Hoc Networks,” *Journal of Communications and Networks,* vol. 3, no. 2, pp. 1–12, June 2001.
117. J. Liu and S. Singh. “ATCP: TCP for Mobile Ad Hoc Networks,” *IEEE Journal on Selected Areas in Communications,* vol. 19, no. 7, pp. 1300-1315, July 2001.
118. J. Liu and S. Singh. “ACTP: Application Controlled Transport Protocol for Mobile Ad Hoc Networks,” In Proc. of *IEEE WCMC*, vol. 3, pp. 1318–1322, September 1999.
119. Z. Fu, X. Meng, and S. Lu. “A Transport Protocol For Supporting Multimedia Streaming in Mobile Ad Hoc Networks,” IEEE JSAC, vol. 21, no. 10, pp. 1615–1626, December 2004.
120. J. W. Hui and D. Culler. “The Dynamic Behavior of a Data Dissemination Protocol for Network Programming at Scale”. In Proc. *The 2nd ACM Conference on Embedded Networked Sensor Systems (SenSys’04)*, November, 2004.
121. K. Sundaresan, V. Anantharaman, H. Y. Hsieh, and R. Sivakumar. “ATP: A Reliable Transport Protocol for Ad Hoc Networks,” In Proc. *ACM MOBIHOC*, pp. 64–75, June 2003.
122. C. Y. Wan, A. T. Campbell and L. Krishnamurthy. “PSFQ: A Reliable Transport Protocol for Wireless Sensor Networks.” In Proc. *WSNA 2002,* September 2002.
123. T. Banka. “PSFQ: A Reliable Transport Protocol for Wireless Sensor Networks.” available: www-net.cs.umass.edu
124. Y. Sankarasubramaniam, . B. Akan, and I. F. Akyildiz. “ESRT: Event to Sink Reliable Transport in Wireless Sensor Networks,” In Proc. MobiHoc03, pp. 177–188, June 2003.

125. Y. G. Iyer, S. Gandham, and S. Venkatesan. “STCP: A Generic Transport Layer Protocol for Wireless Sensor Networks,” In Proc. *14th IEEE International Conference on Computer Communications and Networks,* October 2005.
126. A. Gaddah and T. Kunz. “A Survey of Middleware Paradigms for Mobile Computing”. Carleton University Systems and Computing Engineering Technical Report SCE-03-16, July 2003.
127. R. Handorean, R. Sen, G. Hackmann, and G. Roman. ‘Context Aware Session Management for Services in Ad Hoc Networks,” In Proc. *IEEE International Conference on Services Computing (SCC’05)*, vol 1, pp. 113-120, 2005.
128. T. Plagemann, V. Goebel, C. Griwodz, and P. Halvorsen. “Towards Middleware Services for Mobile Ad-hoc Network Applications”. In Proc. *IEEE Workshop on Future Trends of Distrib uted Computing Systems (FTDCS 2003)*, 2003.
129. W. B. Heinzelman, A. L. Murphy, H. S. Carvalho, and M. A. Perillo. “Middleware to Support Sensor Network Applications”. *IEEE Network Mag.*, 18(1):6–14, 2004.
130. K. Rmer, O. Kasten, and F. Mattern. “Middleware Challenges for Wireless Sensor Networks”. *ACM Mobile Computing and Communication Review,* 6(4): 59–61, October 2002.
131. I. Chlamtac, M. Conti, and J. J. N. Liu. “Mobile ad hoc networking: imperatives and challenges”. *Ad Hoc Networks*, 1:13-64, 2003.
132. M. Musolesi, C, Mascolo, and S. Hailes. “Adapting Asynchronous Messaging Middleware to Ad Hoc Networking”. In Proc. *2nd workshop on Middleware for Pervasive and Ad-Hoc Computing,*, 2004.
133. K. Romer. Programming Paradigms and Middleware for Sensor Networks. *GI/ITG Fachgesprch Sensornetze,* Karlsruhe, February 2004.
134. J. Biswas and S. K. Nandy. “Application Layer Multicasting for Mobile Ad-Hoc Networks with Network Layer Support.” In Proc. *29th Annual IEEE International Conference on Local Computer Networks (LCN04)*, 2004.
135. M. Ge, S. V. Krishnamurthy, and M. Faloutsos “Application versus network layer multicasting in ad hoc networks: the ALMA routing protocol”. *Ad Hoc Networks*, 2004.
136. G.H. Cooper. “The Argument for Soft Layers of Protocol”. Technical Report TR-300, Massachussets Institute of Technology, Cambridge, MA, May 1983.
137. D.D. Clark and D.L. Tennenhouse. “Architectural Considerations for New Generations of Protocols”. In Proc. *ACM SIGCOMM*, Philadelphia, PA, 1990.
138. A. Sanmateu, F. Paint, L. Morand, S. Tessier, P. Fouqart, A. Sollund, and E. Bustos, “Seamless Mobility across IP networks using Mobile IP”. *Computer Networks*, 40(1):181–190, 2002.
139. V. Srivastava and M Motani. “Cross-layer design: a survey and the road ahead.” *IEEE Communications Magazine*, 43(12):112–199, Dec. 2005.
140. Wikipedia. Direct Democracy. available: http://en.wikipedia.org/
141. S. Shakkottai, T. S. Rappaport and P. C. Karlsson. “Cross-layer Design for Wireless Networks,” *IEEE Communications magazine,* October, 2003.
142. S. Floyd. “TCP and Explicit Congestion Notification”. *ACM Computer Communication Review*, vol. 24, pp. 10–23, October, 1994.
143. V. T. Raisinghani and S. Iyer. “Cross-layer Design Optimization in Wireless Protocol Stacks”. *Elsevier Computer Communications*, 27, pp. 720–724, 2003.
144. C. Barrett, M. Drozda, A. Marathe, and M. V. Marathe. “Characterizing the Interaction Between Routing and MAC Protocols in Ad-Hoc Networks”. In Proc. *MOBIHOC’02*, 2002.

145. R. Ludwig, A. Konrad, A.D. Joseph, and R.H. Katz. “Optimizing the End-to-End Performance of Reliable Flows over Wireless Link”. *Wireless Networks*, 8(2/3):289–299, 2002.
146. V. Kawadia and P. R. Kumar. “A Cautionary Perspective on Cross-Layer Design” *IEEE Wireless Communications*, pp. 3–11, Feb. 2005.
147. M. Conti, G. Maselli, G. Turi, and S. Giordano. “Cross-Layering in Mobile Ad Hoc Network Design.” *IEEE Computer*, 37(2):48–51, 2004.
148. R. Winter, J. Schiller, N. Nikaein, and C. Bonnet. “CrossTalk: A Data Dissemination-based Crosslayer Architecture for Mobile Ad-hoc Networks”. In Proc. *Applications and Services in Wireless Networks*, 2005.
149. D. Culler, P. Dutta, C. T. Ee, R. Fonseca, J. Hui, P. Levis, J. Polastre, S. Shenker, I. Stoica, G. Tolle, and J. Zhao. “Towards a Sensor Network Architecture: Lowering the Waistline”. In Proc. *Tenth Workshop on Hot Topics in Operating Systems (HotOS X)*, 2005.
150. P. J. Marron, D. Minder, A. Lachenmann, and K. Rothermel. “TinyCubus: An Adaptive Cross-Layer Framework for Sensor Networks”. *IT - Information Technology*, vol. 47(2), pp. 87–97, 2005.
151. J. Lu, Y. Pan, J. Wang, A. D. Yahaya, and T. Suda. “A Cross-layer Analysis Model for Wireless Sensor Network QoS”. available: netresearch.ics.uci.edu/ariffin/.
152. D. Porcino and W. Hirt. “Ultra-Wideband Radio Technology: Potential and Challenges Ahead”. *IEEE Communications Magazine*, 41:7, 66-74, 2003.
153. S. Stroh. “Ultra-Wide Band: Multimedia Unplugged,” *IEEE Spectrum*, 40:9, 23-27, 2003.
154. J. Gomez, A. T. Campbell, M. Naghshineh, and C. Bisdikian. “Conserving Transmission Power in Wireless Ad Hoc Networks”. In *ICNP*, 2001.
155. S. Agrawal, S. Krishnamurthy, R. H. Katz, and S. K. Dao. “Distributed Power Control in Ad-hoc Wireless Networks”. In *PIMRC*, 2001.
156. E. S. Jung and N. Vaidya. “ Power Control MAC Protocol for Ad Hoc Networks”. In *Mobicom*, 2002.
157. J. Foerster, E. Green, S. Somayazulu, and D. Leeper. “Ultra-Wideband Technology for Short or Medium Range Wireless Communication”. *Intel Technology Journal*, 2001.
158. T. Girici and A. Ephremides. “Joint Routing and Scheduling Metrics in Wireless Ad Hoc Networks”. In Proc. *36th Asilomar Conference on Signals Systems and Computers,* Nov. 2002.
159. R. L. Cruz and A. V. Santhanam. “Optimal routing, link scheduling and power control in multi-hop wireless networks”. IEEE INFOCOM, 2003.
160. T. ElBatt and A. Ephremides. “Joint scheduling and power control for wireless ad hoc networks”. *IEEE Transactions on Wireless Communications*, vol. 1, pp. 74–85, 2004.
161. U. C. Kozat, I. Koutsopoulos, and L. Tassiulas. A framework for cross-layer design of energy-efficient communication with QoS provisioning in multi-hop wireless networks. In Proc. *INFOCOM*, 2004.
162. G. Lu and B. Krishnamachari. “Energy Efficient Joint Scheduling and Power Control in Wireless Sensor Networks”. In Proc. *Second IEEE International Conference on Sensor and Ad hoc Communications and Networks (SECON)*, September 2005.

163. R. Madan, S. Cui, S. Lall, and A. J. Goldsmith. “Cross-Layer Design for Lifetime Maximization in Interference-Limited Wireless Sensor Networks”. In Proc. *IEEE INFOCOM*, March, 2005.
164. S. Cui, R. Madan, A.J. Goldsmith, and S. Lall. “Cross-layer Energy and Delay Optimization in Small-scale Sensor Networks”. Submitted to *IEEE Transactions on Wireless Communications*, 2006.
165. S. Cui, A. J. Goldsmith, and A. Bahai. “Energy-constrained modulation optimization.” *IEEE Trans. on Wireless Communications,* Vol. 4, No. 5, pp. 2349–2360, Sep., 2005.
166. X. Wang and K. Kar. “Cross-Layer Rate Control for End-to-End Proportional Fairness in Wireless Networks with Random Access”. In Proc. *MobiHoc’05*, May, 2005.
167. K. Chen, S. H. Shah and K. Nahrstedt, “Cross-Layer Design for Data Accessibility in Mobile Ad Hoc Networks”. *Kluwer Wireless Personal Communications*, 21: 4976, 2002.
168. J. Polastre, J. Hui, P. Levis, J. Zhao, D. Culler, S. Shenker, and I. Stoica. “A Unifying Link Abstraction for Wireless Sensor Networks”. In Proc. *ACM SenSys’05*, Nov., 2005.
169. P. Levis, N. Patel, D. Culler, and S. Shenker. “Trickle: A self-regulating algorithm for code propagation and maintenance in wireless sensor networks”. In Proc. NSDI 2004, March, 2004.
170. S. Nath, P. B. Gibbons, Z. Anderson, and S. Seshan. “Synopsis diffusion for robust aggregation in sensor networks”. In Proc. *SenSys’04*, Nov., 2004
171. M. L. Sichitiu. “Cross-Layer Scheduling for Power Efficiency in Wireless Sensor Networks”. In Proc. *IEEE Infocom*, 2004.
172. R. Merz, J. Widmer, J.Y. Le Boudec, and B. Radunovic. “A Joint PHY/MAC Architecture for Low Radiated Power TH-UWB Wireless Ad-Hoc Networks” *Wiley Wireless Communications and Mobile Computing Journal*, 5(5), pp. 567–580, 2005.
173. Kay Romer and Friedemann Mattern. “Event-Based Systems for Detecting Real-World States with Sensor Networks: A Critical Analysis”. In Proc. *DEST Workshop on Signal Processing in Sensor Networks at ISSNIP*, pp. 389–395, Melbourne, Australia, December 2004.
174. J. Liu, M. Chu, J. Liu, J. E. Reich, and F. Zhao. “State-centric programming for sensor and actuator network systems”. *IEEE Pervasive Computing Magazine*, vol. 2, no. 4, Oct–Dec. 2003.
175. P. Baldi, L. De Nardis, and M. G. Di Benedetto. “Modelling and Optimization of UWB Communication Networks Through a Flexible Cost Function.” *IEEE Journal on Selected Areas in Communications*, 20(9):1733–1744, 2002.
176. V. Mhatre, C. Rosenberg, D. Kofman, et al. “A Minimum Cost Heterogeneous Sensor Network with a Lifetime Constraint”. Accepted for publication in IEEE Transaction on Mobile Computing, January 2004. available: http://dynamo.ecn.purdue.edu/ cath/tmc04.pdf.
177. Q. Han, S. Mehrotra, and N. Venkatasubramanian. “Energy Efficient Data Collection for Distributed Sensor Environments”. Proceedings of the *24th International Conference on Distributed Computing Systems (ICDCS)*, March 2004.
178. Z. Li and B. Li. “Probabilistic Power Management for Wireless Ad Hoc Networks”. *ACM/Kluwer Mobile Networks and Applications (MONET)*, Special Issue on WLAN Optimization at the MAC and Network Levels, Kluwer Academic Publishers, Vol. 10, No. 5, pp. 771–782, October 2005.

179. R. Jurdak, C. V. Lopes, and P. Baldi. "A Survey, Classification, and Comparative Analysis of Medium Access Control Protocols for Ad Hoc Networks". *IEEE Communications Surveys and Tutorials*, 6:(1), 2004.
180. W. Ye and J. Heidemann. "Medium Access Control in Wireless Sensor Networks". *Wireless Sensor Networks*, Taieb Znati, Krishna M. Sivalingam and Cauligi Raghavendra (eds.), Kluwer Academic Publishers, 2003.
181. Q. Jiang and D. Manivannan. "Routing Protocols for Sensor Networks". In Proc. *IEEE (CCNC)*, 2004.
182. C. E. Perkins and P. Bhagwat. "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers". In Proc. *SIGCOMM 94*, pp. 234–244, 1994.
183. T. Chen and M. Gerla. "Global State Routing: A New Routing Scheme for Ad-hoc Wireless Networks". In Proc. *IEEE ICC*, 1998.
184. C. E. Perkins, E. M. Royer, and S. R. Das. Ad Hoc On-demand Distance Vector Routing. IETF Draft October 99.
185. D. B. Johnson and D. A. Maltz. The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks. IETF Draft October 1999.
186. C. K. Toh. "Maximum battery life routing to support ubiquitous mobile computing in wireless ad hoc networks". *IEEE Comm. Mag.*, pp. 138–147, June 2001.
187. R. C. Shah and J. Rabaey. "Energy Aware Routing for Low Energy Ad Hoc Sensor Networks". In Proc. *IEEE Wireless Communications and Networking Conference (WCNC)*, Orlando, March 2002.
188. R. Szewczyk, et al. "Habitat Monitoring with Sensor Networks". *Comm. of the ACM*, 47(6):34-40, 2004.
189. IEEE 802.15.4. Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs). Standard. available: http://standards.ieee.org/getieee802/download/802.15.4-2003.pdf.
190. W. Ye, J. Heidemann, and D. Estrin. "Medium access control with coordinated, adaptive sleeping for wireless sensor networks". In *IEEE Transactions on Networking*, Apr. 2004.
191. Tiny Operating System. UC Berkeley. available: http://tinyos.net.
192. R. Jurdak, P. Baldi, and C. V. Lopes. "Energy-Aware Adaptive Low Power Listening for Sensor Networks". In Proc. *2nd International Workshop for Networked Sensing Systems (INSS 05)*, San Diego, CA, June 2005.
193. R. Jurdak, P. Baldi, and C. V. Lopes. "State-driven Energy Optimization in Sensor Networks". In Proc. *IEEE/IEE International Conference on Sensor Networks (SENET)*, Montreal, Canada, August 2005.
194. M. Haenggi. "Energy-Balancing Strategies for Wireless Sensor Networks". *IEEE International Symposium on Circuits and Systems (ISCAS'03)*, Bangkok, Thailand, May 2003.
195. F. Zhao and L. Guibas. *Wireless Sensor Networks: An Information Processing Approach.* Morgan Kaufman Publishers, 2004.
196. R. Szewczyk, J. Pollastre, A. Mainwaring, and D. Culler. "Lessons from a Sensor Network Expedition". In Proc *European Workshop on Sensor Networks*, 2004.
197. A. Woo, T. Tong, and D. Culler. "Taming the Underlying Challenges of Reliable Multihop Routing in Sensor Networks". In Proc. *Sensys*, 2003.

198. R. Sivakumar, B. Das, and V. Bharghavan. “The clade vertebrata: spines and routing in ad hoc networks”. In Proc. *Third IEEE Symposium on Computers and Communications (ISCC ’98)*, pp. 599 – 605, 1998.
199. D. Gay, P. Levis, R. von Behren, et al. ”The nesC language: A holistic approach to networked embedded systems”. *ACM SIGPLAN Notices*, 38(5):1–11, 2003.
200. R. Jurdak, P. Baldi, and C. V. Lopes. “U-MAC: A Proactive and Adaptive UWB Medium Access Control Protocol”. *Wiley Wireless Communications and Mobile Computing Journal*, 5(5):551-566, 2005.
201. J. Foerster, E. Green, S. Somayazulu, and D. Leeper. “Ultra-Wideband Technology for Short or Medium Range Wireless Communication”. *Intel Technology Journal*, 2001.
202. D. Porcino and W. Hirt. “Ultra-Wideband Radio Technology: Potential and Challenges Ahead”. *IEEE Communications Magazine*, 41:7, 66-74, 2003.
203. IEEE 802.15.3a . WPAN High Rate Alterantive PHY Task Group 3a. Draft Standard. available: http://www.ieee802.org/15/pub/TG3a.html. accessed: 1/25/05.
204. IEEE 802.15.4a. WPAN Low Rate Alterantive PHY Task Group 4a. Draft Standard. available: http://www.ieee802.org/15/pub/TG4a.html. accessed: 1/25/05.
205. M. Z. Win and R. A. Scholtz. “Ultra-Wide Bandwidth Time-Hopping Spread-Spectrum Impulse Radio for Wireless Multiple-Access Communications”. *IEEE Transactions on Communications*, 48(4):679–691, 2000.
206. F. Cuomo, C. Martello, A. Baiocchi, et al. “Radio resource sharing for ad-hoc networking with UWB”. *IEEE Journal on Selected Areas in Communications*, 20(9):1722–1732, 2002.
207. F. Cuomo, C. Martello and S. Caputo. “An Interference-controlled admission control scheme for QoS support in distributed UWB networks”. In Proc. *IST Mobile and Wireless Communications Summit*, 2003.
208. F. Cuomo and C. Martello. “Improving Wireless Access Control Schemes via Adaptive Power Regulation”. In Proc. *PWC 2003*, pp 114-127, 2003.
209. R. M. Buehrer, D. Ha, J. Reed, D. McKinstry and N. Kumar. “Range Extension for UWB Communications”. Virginia Tech, 2002. available: http://filebox.vt.edu/users/nikumar, Accessed: 1/20/2005.
210. F. Cuomo and C. Martlello. “MAC principles for an ultra wide band wireless access”. In Proc. *Global Telecommunications Conference*, 2001.
211. Whyless.com: The Open Mobile Access Network. available: http://www.whyless.org, Accessed: 1/20/2005.
212. S. Lal and E. S. Sousa. “Distributed Resource Allocation for DS-CDMA-based Multimedia Ad-Hoc Wireless LAN’s”. In *IEEE Journal on Selected Areas in Communications*, 17(5):947-967, 1999.
213. Revision of Part 15 of the Commission’s Rules Regarding Ultra-Wideband Transmission Systems. Federal Communications Commission, ET Docket 98–153, 2002.
214. C-K. Toh. “A Novel Distributed Routing Protocol to Support Ad-hoc Mobile Computing”. In Proc. *IEEE 15th Annu. Int. Phoenix Conf. Computer Communications*, 480-486, 1996.
215. OPNET Modeler software. available: http://www.opnet.com/products/modeler/home.html.
216. S. Stroh. “Ultra-Wide Band: Multimedia Unplugged”. *IEEE Spectrum*, 40:(9), 23–27, 2003.

217. C. E. Smith. "Theory and Design of Directional Antenna Systems". Cleveland Institute of Radio Electronics, 1951.
218. M. Barrett and R. Arnott. "Adaptive Antennas for Mobile Communications". *Electronics and Communication Engineering Journal*, 203–214, 1994.
219. H. G. Schantz. "Introduction to Ultra-Wideband Antennas. Proc. *IEEE Conference on Ultra-Wideband Systems and Technologies*, 2003.
220. M. G. M. Hussain. "Principles of SpaceTime Array Processing for Ultrawide-Band Impulse Radar and Radio Communications". *IEEE Transactions on Vehicular Technology*, 51(3):393–403, 2002.
221. National oceanic and atmospheric administration.
available: http://www.csc.noaa.gov/coos/hawaii.html, 2003.
222. M. Bhardwaj, T. Garnett, and A.P. Chandrakasan. "Upper bounds on the lifetime of sensor networks". In Proc. *International Conference on Communications*, volume 3, pp. 785–790. IEEE, 2001.
223. J. Zhu and Symeon Papavassiliou. "On the Energy-Efficient Organization and the Lifetime of Multi-hop Sensor Networks". *IEEE Communications Letters*, volume 7:11, pp. 537–539, 2003.
224. P.T. Gough and D.W. Hawkins. "A short history of synthetic aperture sonar". In Proc. *IEEE Int.Geoscience and Remote Sens. Symp.*, 1998.
225. P. Chapman, D. Wills, G. Brookes, and P. Stevens. "Visualizing Underwater Environments Using Multi-frequency Sonar". *IEEE Computer Graphics and Applications*, 1999.
226. G.M. Trimble. "Underwater Object Recognition and Automatic Positioning to Support Dynamic Positioning". In Proc *7th Int. Symposium Unmanned Untethered Submersile Technology*, pp. 273–279, 1991.
227. X. Yang, et al. "Design of a Wireless Sensor Network for Longterm, In-Situ Monitoring of an Aqueous Environment". *Sensors*, 2:455-472, 2002.
228. N. Fruehauf and J.A. Rice. "System design aspects of a steerable directional acoustic communications transducer for autonomous undersea systems". In Proc. *OCEANS*, volume 1, pp. 565–573. IEEE, 2000.
229. S. Tilaky, N. B. Abu-Ghazalehy, and W. Heinzelman. "Infrastructure tradeoffs for sensor networks". In Proc. *WSNA*, 2002.
230. M.A. Marsan, C.F. Chiasserini, A. Nucci, G. Carello, and L. De Giovanni. "Optimizing the topology of bluetooth wireless personal area networks". In Proc. *INFOCOM*, volume 2, pp. 572–579. IEEE, 2002.
231. J. H. Chang and L.Tassiulas. "Energy conserving routing in wireless ad-hoc networks". In Proc. *Annual Joint Conference of the IEEE Computer and Communications Societies*, pp. 22-31, 2000.
232. Y. Xu, J. Heidemann, and D. Estrin. "Geography-informed energy conservation for ad hoc routing". In *Mobile Computing and Networking*, pp. 70-84, 2001.
233. E. S. Jung and N. Vaidya. "Power Control MAC Protocol for Ad Hoc Networks". In Proc.*Mobicom*, 2002.
234. S. Agrawal, S. Krishnamurthy, R. H. Katz, and S. K. Dao. "Distributed Power Control in Ad-hoc Wireless Networks". In Proc.*PIMRC*, 2001.
235. D. Panigrahi, C. Chiasserini, S. Dey, R. Rao, A. Raghunathan, and K. Lahiri. "Battery life estimation of mobile embedded systems." In Proc. *Fourteenth International Conference on VLSI Design*, pp. 57 –63, 2001.
236. Underwater Acoustic Modem. available: www.link-quest.com.

237. K. Kalpakis, K. Dasgupta, and P. Namjoshi. "Maximum lifetime data gathering and aggregation in wireless sensor networks". In Proc.*International Conference on Networking*, pp. 685–696, IEEE, 2002.

238. J. G. Proakis, J.A. Rice, and M. Stojanovic. "Shallow water acoustic networks". *IEEE Communications Magazine*, (11), 2001.

239. M. Stojanovic. "Recent advances in high speed underwater acoustic communications". *Oceanic Engineering*, 21(4):125–36, 1996.

240. S. A. L. Glegg, R. Pirie, and A. LaVigne. A study of ambient noise in shallow water, available: http://www.oe.fau.edu/acoustics/.

241. A. B. Boehm, S. B. Grant, J. H. Kim, S. L. Mowbray, C. D. McGee, C. D. Clark, D. M. Foley, and D. E. Wellman. "Decadal and shorter period variability of surf zone water quality at huntington beach, california". *Environmental Science and Technology*, 2000.

242. R. Holman, J. Stanley and T. Ozkan-Haller. "Applying Video Sensor Networks to Nearshore Environment Monitoring". *Pervasive Computing*, pp. 14–21, 2003.

243. Z. Xinhua. "An information model and method of feature fusion". In Proc. *ICSP*, 1998.

244. F. H. Fisher and V. P. Simmons. " Sound Absorption in Sea Water". *Journal of Acoustical Society of America*, 62:558, 1977.

245. A. Misra and S. Banerjee. "MPRC: Maximizing Network Lifetime for Reliable Routing in Wireless Environments". In Proc. *IEEE WCNC*, 2002.

246. F. H. Fisher. "Effect of High Pressure on Sound Absorption and Chemical Equilibrium". *Journal of Acoustical Society of America*, 30:442, 1958.

247. R. Jurdak, C. V. Lopes, and P. Baldi. "Programming Model for Supporting Complex Optimizations in Sensor Networks". In Proc. *Fourth International IEEE/ACM Conference on Information Processing in Sensor Networks (IPSN05)* Work-in-Progress. Los Angeles, CA, 2005.

Index

SIGNALS AND COMMUNICATION TECHNOLOGY

(continued from page ii)

Speech Enhancement
J. Benesty (Ed.)
ISBN 3-540-24039-X

Multimedia Communication Technology
Representation, Transmission
and Identification of Multimedia Signals
J.R. Ohm ISBN 3-540-01249-4

Information Measures
Information and its Description in Science
and Engineering
C. Arndt ISBN 3-540-40855-X

Processing of SAR Data
Fundamentals, Signal Processing,
Interferometry
A. Hein ISBN 3-540-05043-4

Chaos-Based Digital Communication Systems
Operating Principles, Analysis Methods, and
Performance Evalutation
F.C.M. Lau and C.K. Tse
ISBN 3-540-00602-8

Adaptive Signal Processing
Application to Real-World Problems
J. Benesty and Y. Huang (Eds.)
ISBN 3-540-00051-8

Multimedia Information Retrieval and Management Technological
Fundamentals and Applications D. Feng, W.C.
Siu, and H.J. Zhang (Eds.)
ISBN 3-540-00244-8

Structured Cable Systems
A.B. Semenov, S.K. Strizhakov,and I.R.
Suncheley
ISBN 3-540-43000-8

UMTS
The Physical Layer of the Universal Mobile
Telecommunications System
A. Springer and R. Weigel
ISBN 3-540-42162-9

Advanced Theory of Signal Detection
Weak Signal Detection in Generalized
Obeservations
I. Song, J. Bae, and S.Y. Kim
ISBN 3-540-43064-4

Wireless Internet Access over GSMand UMTS
M. Taferner and E. Bonek
ISBN 3-540-42551-9

Printed in the United States of America.