Signals and Communication Technology

Habib M. Ammari *Editor*

The Art of Wireless Sensor Networks

Volume 2: Advanced Topics and Applications

Springer

Signals and Communication Technology

For further volumes:
http://www.springer.com/series/4748

Habib M. Ammari
Editor

The Art of Wireless Sensor Networks

Volume 2: Advanced Topics and Applications

Springer

Editor
Habib M. Ammari
WiSeMAN Research Lab
Department of Computer
and Information Science
College of Engineering
and Computer Science
University of Michigan-Dearborn
Dearborn, MI
USA

ISSN 1860-4862 ISSN 1860-4870 (electronic)
ISBN 978-3-642-40065-0 ISBN 978-3-642-40066-7 (eBook)
DOI 10.1007/978-3-642-40066-7
Springer Heidelberg New York Dordrecht London

Library of Congress Control Number: 2013953605

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

To my first teachers: My mother, Mbarka, and my father, Mokhtar

To my very best friends: My wife, Fadhila, and my children, Leena, Muath, Mohamed-Eyed, Lama, and Maitham

To my first Dean: Dr. Bernard J. Firestone Professor of Political Sciences and Dean of the Hofstra College of Liberal Arts and Sciences at Hofstra University for his wonderful friendship and outstanding support to me during my stay at Hofstra University (Sep 2008–Aug 2011)

To my second Dean: Dr. Subrata Sengupta Professor of Mechanical Engineering and Dean of the College of Engineering and Computer Science at the University of Michigan-Dearborn for his wonderful friendship and outstanding support to me

Foreword

Over close to two decades of research, wireless sensor networks have transformed from a bold vision to ever-expanding reality. While the original driving force, hardware miniaturization and integrated sensing, computing and communication, remains critical, the variety of sensor network applications that such technologies have enabled are likely beyond earlier imaginations. With more and more experience from case studies and real-world deployments, the field is simultaneously maturing and evolving, addressing new challenges in both theory and practice. Through this process, sensor network research has become a truly interdisciplinary field, cutting across areas such as embedded systems, operating systems, signal processing, communication theory, networking, and computational geometry, among many others.

This two-volume book is therefore a very timely re-examination of this field. This second volume, entitled "*The Art of Wireless Sensor Networks: Advanced Topics and Applications*", presents a collection of recent advances in both theory and application of wireless sensor networks. Specifically, it starts with a stochastic modeling framework aimed at better understanding and evaluating key QoS metrics of a wireless sensor network, such as packet delivery delay, energy consumption, and lifetime. This is followed by a collection of chapters that discuss the state-of-the-art in various theoretical aspects and enabling technologies critical to sensor networking. These chapters build on an increasing volume of literature but also introduce new concepts. They are thus not only wonderful references but also good sources of new ideas. These include advanced topics in sensing coverage (barrier, spatial and temporal), indoor tracking, real-time estimation, and target counting. It also includes a set of chapters dedicated to 3D sensor networks and research challenges that arise in coverage, connectivity, localization, topology control, and routing. The Part III of this volume consists of chapters that discuss emerging applications that are beginning to have significant scientific and societal impact. These include underground sensing, underwater sensing, multimedia sensing, and body area sensing (its analysis as well as its use in activity and gesture recognition). This part ends with a few interesting chapters on social sensing: the sensing of, for, and within social/human networks.

With a field so diverse as evidenced by this collection of chapters, it has been an enormous effort, but a decidedly worthwhile one, to put together these two volumes. Much of the credit goes to the editor of the book, Dr. Habib M. Ammari.

These two volumes present a glimpse into many exciting and vibrant research directions within the field of wireless sensor networks, and should find an interested audience in both practitioners and theoreticians, and in novices and experts alike.

March 10, 2013

Prof. Mingyan Liu
University of Michigan
Ann Arbor
MI, USA

Contents

Part VIII Social Sensing

Contributors

Tarek F. Abdelzaher University of Illinois Urbana-Champaign, Urbana, USA

Habib M. Ammari University of Michigan-Dearborn, Dearborn, MI, USA

Jeffrey A. Burke University of California Los Angeles, Los Angeles, USA

Baozhi Chen Rutgers University, New Brunswick, USA

Xiuzhen Cheng George Washington University, Washington, DC, USA

Yaser P. Fallah West Virginia University, Morgantown, USA

Xinwen Fu University of Massachusetts Lowell, Lowell, USA

Steve Goddard University of Nebraska-Lincoln, Lincoln, USA

Dong Guo Boston University, Boston, USA

Ahmed Helmy University of Florida, Gainesville, USA

Wei-Jen Hsu University of Florida, Gainesville, USA

Narayanan Krishnan Washington State University, Pullman, USA

Hongjuan Li George Washington University, Washington, DC, USA

Keyong Li Boston University, Boston, USA

Yingwei Lin Boston University, Boston, USA

Benyuan Liu University of Massachusetts Lowell, Lowell, USA

Mingyan Liu University of Michigan, Ann Arbor, USA

Tommaso Melodia State University of New York at Buffalo, Buffalo, USA

Mahta Moghaddam University of Southern California, Los Angeles, USA

Sethuraman Panchanathan Arizona State University, Phoenix, USA

Ioannis Ch. Paschalidis Boston University, Boston, USA

Dario Pompili Rutgers University, New Brunswick, USA

Scott Pudlewski Massachusetts Institute of Technology, Cambridge, USA

Raja Sengupta University of California Berkeley, Berkeley, USA

Anwar Saipulla University of Massachusetts Lowell, Lowell, USA

Agnelo Rocha da Silva University of Southern California, Los Angeles, USA

Rui Tan Advanced Digital Sciences Center, Singapore, Singapore

Mehmet C. Vuran University of Nebraska-Lincoln, Lincoln, USA

Dong Wang University of Illinois Urbana-Champaign, Urbana, USA

Jie Wang University of Massachusetts Lowell, Lowell, USA

Yu Wang University of North Carolina at Charlotte, Charlotte, USA

Yunbo Wang University of Nebraska-Lincoln, Lincoln, USA

Dengyuan Wu George Washington University, Washington, DC, USA

Guoliang Xing Michigan State University, East Lansing, USA

Bowu Zhang George Washington University, Washington, DC, USA

Part I
Introduction and Stochastic Modeling

Chapter 1
Introduction

Habib M. Ammari

"I have tried to write this set of books in such a way that it will fill several needs. In the first place, these books are reference works that summarize the knowledge that has been acquired in several important fields. In the second place, they can be used as textbooks for self-study or for college courses in the computer and information sciences"

Donald E. Knuth, *The Art of Computer Programming: Fundamental Algorithms* (1997)

1 The Art of Wireless Sensor Networks

Nowadays, the design and development of wireless sensor networks for various real-world applications, such as environmental monitoring, health monitoring, industrial process automation, battlefields surveillance, and seism monitoring, has become possible owing to the rapid advances in both of wireless communications and sensor technology. This type of network is cost-effective and appealing to a wide range of mission-critical situations. These two reasons helped them gain significant popularity compared to other types of networks. A wireless sensor network is a collection of low-powered, physically tiny devices, called *sensor nodes*, which are capable of sensing the physical environment, collecting and processing sensed data, and communicating with each other in order to accomplish certain common tasks. Furthermore, wireless sensor networks possess a central gathering point, called the *sink* (or *base station*), where all the collected data can be stored. The major challenge in the design and development of wireless sensor networks is mainly due to the severe

H. M. Ammari (✉)
WiSeMAN Research Lab, University of Michigan-Dearborn, Dearborn, MI, USA
e-mail: hammari@umd.umich.edu

H. M. Ammari (ed.), *The Art of Wireless Sensor Networks*,
Signals and Communication Technology, DOI: 10.1007/978-3-642-40066-7_1,

constraints that are imposed on the sensing, storage, processing, and communication features of the sensor nodes. More precisely, the sensor nodes suffer from severely constrained power supplies, which shorten their lifetime and make them unreliable. It is worth noting that the sensor nodes may become faulty due to improper hardware functioning and/or low battery power (or energy). The latter is very crucial to be considered in the design and implementation of this type of network for their correct operation and longevity.

Since their inception in the late 1990s, wireless sensor networks have witnessed significant growth and tremendous development in both academia and industry. A large number of researchers, including computer scientists and engineers, have been interested in solving challenging problems that span all the layers of the protocol stack of sensor networking systems. Several venues, such as journals, conferences, and workshops, have been launched to cover innovative research and practice in this promising and rapidly advancing field. Because of these trends, I thought it would be beneficial to provide our sensor networks community with a comprehensive reference on as much of the findings as possible on a variety of topics in wireless sensor networks. As this area of research is in continuous progress, it does not seem to be a reasonable solution to keep delaying the publication of such reference any more.

This book series, titled "*The Art of Wireless Sensor Networks*," has two volumes that have been designed in a way to address challenging problems in traditional as well as new emerging areas of research in sensor networking. Moreover, all the book chapters in both volumes have been written as surveys of the state-of-the-art and state-of-the-practice of their corresponding topics. Our main goal is to help the readers understand the basic concepts of wireless sensor networks, and also be aware and knowledgeable of most of the underlying research topics although some of them are still in their infancy and not much work has been done to solve those new research problems. These two volumes are titled:

- *The Art of Wireless Sensor Networks: Fundamentals*
- *The Art of Wireless Sensor Networks: Advanced Topics and Applications*

This book relates to the second volume and focuses on the advanced topics and applications of wireless sensor networks. Based on my fruitful discussion with the contributing authors whom I invited, and, particularly, Drs. Wendi Heinzelman, Kay Römer, and Mohamed Younis, our rationale is that the second volume has all application-specific and non-conventional sensor networks, emerging techniques, and advanced topics that are not as matured as what is covered in the first volume. Thus, the second volume deals with three-dimensional, underground, underwater, body-mounted, and societal networks. Following Donald E. Knuth's above-quoted elegant strategy to focus on several important fields (*The Art of Computer Programming: Fundamental Algorithms*, 1997), all the book chapters in this volume include up-to-date research work spanning various topics, such as stochastic modeling, barrier and spatiotemporal coverage, tracking, estimation, counting, coverage and localization in three-dimensional sensor networks, topology control and routing in three-dimensional sensor networks, underground and underwater sensor networks, multimedia and body sensor networks, and social sensing. Most of these major topics

can be covered in an advanced course on wireless sensor networks. This book will be an excellent source of information for graduate students majoring in computer science, computer engineering, electrical engineering, or any related discipline. Furthermore, computer scientists, researchers, and practitioners in both academia and industry will find this book useful and interesting.

I would like to mention that I borrowed the title of this two-volume book series, "*The Art of Wireless Sensor Networks*," from Dr. Donald E. Knuth, computer scientist and Professor Emeritus at Stanford University, who is the author of the seminal multi-volume set of books, titled "*The Art of Computer Programming*." In fact, most of the problems being addressed in the area of wireless sensor networks are challenging and mathematical in nature. And, solving those problems requires an 'art' to find elegant yet efficient solutions in terms of time, space, and, especially, energy, which is a crucial resource in the design and implementation of algorithms and protocols for wireless sensor networking systems. I hope the readers will see the 'art' in this book and enjoy reading it as much as I enjoyed editing it.

2 Book Organization

This book has eight parts, each of which includes 2–3 chapters. Next, we briefly summarize the purpose of each part with a short description of its chapters.

In Part 1, titled "*Introduction and Stochastic Modeling*," Chap. 2 provides a comprehensive cross-layer probabilistic analysis framework in order to investigate the probabilistic evaluation of quality of service performance provided by wireless sensor networks. The latter is evaluated in two levels, namely node level and network level.

In Part 2, titled "*Barrier and Spatiotemporal Coverage*," Chap. 3 presents a comprehensive survey on barrier coverage of wireless sensor networks. It focuses on the critical conditions and construction of barrier coverage in 2-dimensional wireless sensor networks, barrier coverage under a line-based sensor deployment scheme, as well as the effect of sensor mobility on barrier coverage, and barrier coverage in three-dimensional underwater sensor networks. Chapter 4 explores the fundamental limits of spatiotemporal coverage based on stochastic data fusion models in order to capture the stochastic nature of sensing. It derives the scaling laws between spatiotemporal coverage, network density, and signal-to-noise ratio. Also, it shows that data fusion can significantly improve spatiotemporal coverage by exploiting the collaboration among sensors when several physical properties of the target signal are known.

In Part 3, titled "*Tracking, Estimation, and Counting*," Chap. 5 describes various methods using sophisticated computations in pursuit of high localization accuracy with low hardware investment and moderate set-up cost. Also, it shows a profile-based approach to infer the positions of mobile wireless devices in complex indoor environments, a two-tier statistical positioning scheme that helps improve efficiency by adding movement detection, and joint cluster-head placement optimization for

localization and movement detection. Chapter 6 discusses methods and protocols for controlling the behavior of nodes in order to allow maximal use of the shared medium for real-time estimation. It shows how transmission control protocols can be applied in a wireless network of mobile sensors to achieve high accuracy given the limitations of the medium. Chapter 7 reviews four existing classes of counting methods, namely binary counting, numeric counting, energy counting, and compressive counting. It describes methods for each class and discusses their advantages and disadvantages. Moreover, it compares those methods to illustrate the impact of different sensor network settings on the target counting accuracy.

In Part 4, titled "*Coverage and Localization in Three-Dimensional Wireless Sensor Networks*," Chap. 8 surveys existing methods on coverage and connectivity in three-dimensional wireless sensor networks. It provides a study of different placement strategies, fundamental characteristics, modeling schemes, analytical methods, and limiting factors, as well as the practical constraints imposed on coverage and connectivity in three-dimensional wireless sensor networks. Chapter 9 focuses on localization in three-dimensional wireless sensor networks. It describes generic, airborne, terrestrial, and submerged localization schemes along with their strengths and weaknesses.

In Part 5, titled "*Topology Control and Routing in Three-Dimensional Wireless Sensor Networks*," Chap. 10 presents most recent work on three-dimensional topology control. Chapter 11 reviews several classes of routing techniques in three-dimensional wireless sensor networks.

In Part 6, titled "*Underground and Underwater Sensor Networks*," Chap. 12 discusses the challenges facing the design of underground sensor networks, the perceived limitations, and recent technological advances in this field. It shows that the design of underground sensor networks must be tailored to the application. As an illustrative example, a proposed study concludes that the design of an underground sensor network for detecting oil pipeline leakage is totally different from that of an underground sensor network for agricultural draught or landslide monitoring. Chapter 13 presents a solution to optimize communications in autonomous underwater vehicles by delaying packet transmissions while waiting for a favorable network topology.

In Part 7, titled "*Multimedia and Body Sensor Networks*," Chap. 14 examines the challenges in the implementation of wireless multimedia sensor network, and how to develop one with the same performance as a traditional scalar wireless sensor network. Also, it shows how to exploit compressed sensing to reduce the energy consumption due to encoding and transmitting high quality video in a severely resource constrained environment. Chapter 15 considers one application of body sensor networks, which involves processing of wearable accelerometer data for recognizing ambulatory or simple activities and activity gestures. Also, it discusses various aspects that are associated with a real-time activity recognition system.

In Part 8, titled "*Social Sensing*," Chap. 16 describes new research challenges that need to be addressed in social sensing frameworks, which should allow massive information dissemination. Chapter 17 reviews existing work on behavior-aware routing. Also, it presents a framework showing all the steps involved in the design

of social behavior-aware routing. Chapter 18 gives an overview of cultural applications of wireless sensing systems from four perspectives, namely the Internet of things, smart grid, participatory sensing, and event-based point of view. Moreover, it discusses the challenges and unique requirements of these applications.

3 Acknowledgments

This book of this complete two-volume series, titled "*The Art of Wireless Sensor Networks*," is a tribute to the fine work of the foremost leading authorities and scholars in their fields of research in the area of sensor networking. Frankly, it is not fair that I am the only one whose name appears on the book cover. And, it is a great pleasure and an honor for me to cordially recognize all of those who contributed a lot to this book and generously supported me throughout this project in order to make this two-volume series a reality. Without them, it would not be possible at all to finish this book and make it available to all the researchers and practitioners, who are interested in the advanced topics and applications of wireless sensor networks.

First and foremost, I am sincerely and permanently grateful to Allah—the Most Gracious, the Most Merciful—for everything He has been providing me with. Particularly, I would very much love to thank Him for giving me the golden opportunity to work with such group of outstanding scientists and researchers to put together this book, and for helping me publish it within 2 years. I am very pleased to dedicate this modest book to Him and very much hope that He would kindly accept it and put His Blessing in it. His Saying "**And of knowledge, you (mankind) have been given only a *little***" has an endless, pleasant echo in my heart and always reminds me that our knowledge is much less than a drop in the ocean.

It is worth mentioning that all the contributing authors were invited to contribute to this book, and that no Call for Book Chapters had ever been sent through any mailing list. All of those authors whom I invited were chosen very selectively to cover most of the advanced topics and applications of wireless sensor networks. They have been contributing to the growth and development of the field of wireless sensor networks. This book would never have been written without their great contributions, support, and cooperation. Therefore, my cordial recognition is due to my colleagues—the ones whom I invited to contribute with their book chapters to this book—whose names are listed in the alphabetical order: Drs. Tarek F. Abdelzaher, Jeffrey A. Burke, Xiuzhen Cheng, Ahmed Helmy, Benyuan Liu, Mingyan Liu, Tommaso Melodia, Sethuraman Panchanathan, Yannis Paschalidis, Dario Pompili, Raja Sengupta, Mehmet Can Vuran, Yu Wang, and Guoliang Xing. I am really honored to have led and worked with such an amazing crew of scientists. I learned a lot from them throughout this project, and it was an incredible experience for me in finishing this book.

Every book chapter has undergone two rounds of reviews. Moreover, in each round, every book chapter received 3–5 reviews by experts in the scope of the chapter. Our ultimate goal is to provide the readers with a high-quality reference on the advanced topics and applications of wireless sensor networks. Precisely, all book chapters were carefully reviewed in both rounds by all the contributing authors.

I would like to express my sincere gratitude to all the contributing authors for their constructive feedback to improve the organization and content of all book chapters. My special thanks go to Dr. Stephan Olariu for his generous offer to review all book chapters of both books of this two-volume series. Also, my original plan was to publish only one book, titled "*The Art of Wireless Sensor Networks*". But, I ended up with 40 book chapters. Therefore, I suggested to all the above-mentioned invited authors to split the book (i.e., 40 book chapters) into two volumes along with their book chapters and titles. Here, again, my special thanks go to all the invited authors for their very helpful feedback with regard to the content of each volume. Moreover, I am very grateful to Dr. Mingyan Liu, Professor of Electrical Engineering and Computer Science at the University of Michigan, Ann Arbor, for her great foreword.

I started this project on Sunday, August 28, 2011 at 06:56 AM when I contacted the Publishing Editor, Dr. Thomas Ditzinger, who approved my proposal for an edited book. All book chapters for both volumes were uploaded on the website of Springer and made accessible to the Editorial Assistant, Mr. Holger Schaepe, on March 11, 2013. Hence, this project lasted over 18 months. During all this period of time, I exchanged 4,840 emails with all contributing authors with regard to their book chapters. I would like to thank all the contributing authors for their invaluable time, flexibility, and wonderful patience in responding to all of my emails in a timely manner. Please forgive me for your time, and I hope that the readers will appreciate all of your great efforts and love all the materials in this book. We all have devoted a considerable time to finish this book and hope it will be paid off in the future.

I would like to acknowledge my family members who have provided me with excellent source of support and constant encouragement over the course of this project. In particular, I am most grateful to my best friend and beloved wife, Fadhila, for her genuine friendship and good sense of humor, and for being extremely supportive and unboundedly patient while I was working on this book. My special thanks and deep appreciation go to her for putting the *Art* into this book. In addition, I would like to express my hearty gratitude to my lovely and beautiful children, Leena, Muath, Mohamed-Eyed, Lama, and Maitham, for their endless support and encouragement. They have been one of my greatest joys, very patient, and understanding. I hope they will forgive me for spending several hours away from them while I was setting in front of my PC in my office or my laptop at home busy with this book. Several times, they all told me: "Daddy, your books and emails are always dragging you away from us!" My lovely wife and children have been a wonderful inspiration to me, and very patient throughout the life of this project. Without their warm love and care, this project would never even have been started. Also, I owe a lot to both of my first teachers, my mother, Mbarka, and my father, Mokhtar, for their sincere prayers, love, support, and encouragement, and for always teaching me and reminding me of the value of knowledge and the importance of family. Furthermore, I would like to thank my mother-in-law, Hania, and my father-in-law, Hedi, for their thoughtful prayers, concern, and valuable support. Besides, I would like to thank my sisters, sisters-in-law, brother, and brothers-in-law for their support and thoughts.

This project could not have been completed without the great support of the people around me who made this experience successful and more than enjoyable.

I would like to thank all of my colleagues and friends at the University of Michigan-Dearborn and, particularly, the four departments of the College of Engineering and Computer Science (CECS), namely Department of Computer and Information Science, Department of Electrical and Computer Engineering, Department of Industrial and Manufacturing Systems Engineering, and Department of Mechanical Engineering, for the collegial and very friendly atmosphere they provided me with to finish this book. In particular, I am very grateful to Dr. Subrata Sengupta, Professor of Mechanical Engineering and CECS Dean at the University of Michigan-Dearborn, for his kindness, continuous encouragement, and outstanding support to WiSeMAN Research Lab since I joined the Department of Computer and Information Science at the University of Michigan-Dearborn on September 1, 2011. Also, I am very thankful to my colleagues at the University of Michigan-Dearborn, namely Dr. Kiumi Akingbehin, Professor of Computer and Information Science; Dr. Yubao Chen, Professor of Industrial and Manufacturing Systems Engineering, and China Programs Director; Dr. Bruce Elenbogen, Associate Professor of Computer and Information Science; Dr. Brahim Medjahed, Associate Professor of Computer and Information Science; Dr. Chris Mi, Professor of Electrical and Computer Engineering, and IEEE Fellow; Dr. Yi Lu Murphey, Professor and Chair of the Department of Electrical and Computer Engineering, and IEEE Fellow; Dr. Elsayed Orady, Professor of Industrial and Manufacturing Systems Engineering; Dr. Adnan Shaout, Professor of Electrical and Computer Engineering; Dr. Paul Watta, Associate Professor of Electrical and Computer Engineering; Dr. David Yoon, Associate Professor of Computer and Information Science; Dr. Armen Zakarian, Professor and Chair of the Department of Industrial and Manufacturing Systems Engineering; and Dr. Qiang Zhu, Professor of Computer and Information Science, and ACM Distinguished Scientist; for their wonderful friendship and for being so supportive and helpful along the way. In addition, I would like to express my special thanks and gratitude to my colleague, Dr. Drew B. Buchanan, Director of the Office of Research and Sponsored Programs, for his excellent support to my research activities in several ways. This work is partially supported by the National Science Foundation (NSF) grants 0917089 and 1054935.

Last, but not the least, I would like to express my deep appreciation to Dr. Thomas Ditzinger, Publishing Editor, Dr. Dieter Merkle, Editorial Director, Mr. Holger Schaepe, Editorial Assistant, Springer-Verlag, Heidelberg (Germany) and New York (USA), Ms. Jacqueline Lenz, Springer Production Manager Book Production STM Heidelberg, Ms. Ramyakrishnan Murugesan, Springer Production Editor, Dr. S.A. Shine David, Project Manager, and Mr. Srinivas and his typesetting team, for their assistance throughout the lifecycle of this project, Varddhene V., Scientific Publishing Services, and Ms. Jessica Wengrzik, Springer DE. It was a great pleasure to work with all of them. I would like to acknowledge the publisher, Springer, for the professionalism and the high quality of their typesetting team as well as their timely publication of this book.

June 2013

Habib M. Ammari
WiSeMAN Research Lab

Chapter 2
Stochastic Modeling of Delay, Energy Consumption, and Lifetime

Yunbo Wang, Mehmet C. Vuran and Steve Goddard

Abstract Emerging applications of wireless sensor networks (WSNs) require real-time quality of service (QoS) guarantees to be provided by the network. Due to the non-deterministic impacts of the wireless channel and queuing state, probabilistic analysis of QoS is essential. For most WSNs applications, the end-to-end delay for packet delivery and the energy consumption are the most important QoS metrics. In this chapter, a comprehensive cross-layer probabilistic analysis framework is presented to investigate the probabilistic evaluation of QoS performance provided by WSNs. In particular, the QoS performance is evaluated in two levels. In the node level, using a Discrete-Time Markov queueing model, the distribution of single-hop delay and single-node energy consumption and lifetime are analyzed. In the network level, based on the node level analysis, the distributions of end-to-end delay, the network lifetime, and the event detection delay are then analyzed. Fluid models are utilized in the network level analysis. The framework also considers a realistic channel environments. Compared to the first-order QoS statistics, such as the mean and the variance, the distribution of QoS metrics reveals the relationship between the performance and reliability with QoS-based operations in WSNs. Using the framework, effective network development can be performed.

Y. Wang (✉) · M. C. Vuran · S. Goddard
University of Nebraska-Lincoln, Lincoln, NE, USA
e-mail: yunbow@cse.unl.edu

M. C. Vuran
e-mail: mcvuran@cse.unl.edu

S. Goddard
e-mail: goddard@cse.unl.edu

H. M. Ammari (ed.), *The Art of Wireless Sensor Networks*,
Signals and Communication Technology, DOI: 10.1007/978-3-642-40066-7_2,

1 Introduction

Wireless sensor networks (WSNs) have been utilized in many applications as both a connectivity infrastructure and a distributed data generation network due to their ubiquitous and flexible nature [4]. Increasingly, a large number of WSN applications are investigated with various quality requirements for different network services specific to low-cost hardware, and unpredictable environment conditions [3, 12]. These requirements necessitate a comprehensive analysis of the Quality of Service (QoS) provided by the network.

QoS issues and techniques have been intensively investigated for ATM networks [14, 45], IP networks [5, 44, 45], and traditional wireless networks [11, 47]. In these studies, the evaluation of QoS is mainly focused on the communication quality characterized by communication delay, jitter, bandwidth, and loss rate. Traditional metrics, however, cannot fully characterize the QoS in WSNs [12], because of the distinct characteristics of WSN applications.

WSNs are utilized for a different set of applications from those with traditional networks [4]. These applications emphasize different characteristics of the network and require different *services* to be provided by the network. Thus, the metrics to evaluate the *quality* of these services are also different from traditional QoS metrics. For example, for most WSN applications, sensor nodes are powered by batteries with limited capacity, and replacing the batteries is difficult. Thus, the network lifetime under battery constraints is a QoS measure that is more important than in traditional network analysis. Other examples of such QoS measures include the delay for event detection, and sensing rate of individual sensors.

Due to limited resource availability, QoS analysis must be performed in a cross-layer manner. In traditional network analysis, with adequate resources assumed, the maintainability and modularity are emphasized at the expense of additionally consumed resources such as storage, computing power, and energy supply. Hence, the QoS is separately provided by different network layers. In contrast, with limited resources, WSNs are usually designed to exploit cross-layer operations and meet QoS requirements more efficiently [46]. For example, cross-layer integration can lead to significant energy conservation [65, 66]. Moreover, requirements on different QoS metrics can contradict with each other, and a tradeoff must be made to provide optimal services. For example, lower delay and longer lifetime are usually contradicting design goals. Lower delay usually requires a high duty cycle, whereas longer lifetime usually favors low duty cycle. Therefore, a QoS analysis framework that captures the tradeoffs for the protocols and operations in the entire software stack is desirable.

1.1 Why Stochastic Models?

In this chapter, we discuss stochastic analysis of QoS provision in WSNs. Stochastic models are necessary for WSNs due to three main factors that result in random operation characteristics.

- Environmental Conditions: WSNs encounter environment conditions that are unreliable and random in nature. Many applications in harsh environments such as wild fields and battlegrounds further impose possible physical damage to the nodes [4]. Any degradation in the performance of one of the nodes would result in unpredicted fluctuations in network performance.
- Low-cost Hardware: Sensor nodes are usually manufactured *en masse* with low-cost hardware. Thus, it is expected that the nodes may randomly cease to work, resulting in a random network topology.
- Wireless Channel: The wireless communication among nodes are also prone to random noises due to low-profile radio transceivers and limited communication power as well as wireless channel impurities such as multi-path fading and shadowing.

All these random factors result in a large variance in QoS metrics, and cannot be thoroughly evaluated using traditional approaches, such as mean delay analysis [1, 7, 28], or worst-case analysis [9, 22].

As wireless sensor network (WSN) technology has matured, more demanding applications have emerged. These applications require quality of service (QoS) guarantees with respect to end-to-end delay, energy consumption, lifetime, and throughput, with high confidence. One example is a smart space application, where resource-constrained wireless sensors and actuators provide situational awareness and assistance to disabled or elderly people. Therein, a random and a rare event, such as a person falling, should be detected within a bounded delay, while still maintaining acceptable throughput and energy consumption levels.

Designing such systems, however, is challenging due to their stochastic characteristics. The common practice of optimizing QoS metrics based on first-order statistics (e.g., mean) is convenient but may be insufficient—especially when the cumulative distribution function (*cdf*) of the QoS metric is not based on a Gaussian distribution. In the case of packet delay, for example, this technique results in good average-case performance, but some packets may experience delays 20 times longer than the average [69, 71]. The problem becomes significantly harder when one considers multiple QoS metrics simultaneously.

We envision that the utilization of these *stochastic analysis techniques* will transform the network design approaches from utility-based solutions to stochastic design principles. In other words, the developed tools are expected to provide the necessary *knobs* that can be tuned to satisfy system requirements and the principles on *how* to tune them.

1.2 Chapter Overview

The remainder of the chapter is organized as follows: In Sect. 2, an overview of the anycast protocol used for illustration of the models is provided. Then, in Sect. 3, stochastic analysis of delay in WSNs is discussed. Following the model described in

this section, the event detection delay distribution is analyzed in Sect. 4. Finally, the stochastic distribution of energy consumption and lifetime in WSNs is presented in Sect. 5. As a whole, this chapter provides a comprehensive tool set for the stochastic analysis of QoS provision in WSNs and presents validations of these analysis tools through simulations and testbed experiments.

2 Anycast Protocol

To illustrate the practical applicability of the analysis tools, anycast protocol is used as a case study throughout the chapter. Other protocols can easily be used following the techniques illustrated for the anycast protocol. The derivations for other protocols (e.g., TDMA, geographical routing, etc.) are left to the reader.

Recent protocol developed for WSNs employ duty cycle operations to save communication energy [8, 57]. To combat the delay incurred due to this duty cycle operation, opportunistic routing techniques, particularly anycast protocols, are utilized along with a high node density to exploit node deployment redundancy [33, 35, 41, 55, 66]. The anycast technique is a cross-layer approach that exploits both temporal and spatial efficiency.

With the anycast technique, if a node has packets to send, it first broadcasts a series of beacon messages. Then, one of the responding neighbors is chosen as the next-hop node according to predefined rules (e.g., the first node that responds, or the closest node to the destination). Finally, the sender forwards the data packet to the chosen neighbor. There are several variations of this basic anycast technique in WSNs. In this chapter, we consider the following representative protocol as an example to illustrate the stochastic analysis techniques.

The anycast protocol operation is illustrated in Fig. 1. Sensor nodes report their readings to the sink through multi-hop routes in the network. The nodes (excluding the sink) turn off their radio periodically to save energy. When a node $\boldsymbol{x}$ has a packet to send it starts to repeatedly transmit Request-to-Send (RTS) beacon packets. These packets are sent using a channel sensing mechanism before the beacon transmission. As shown in Fig. 1, when any other node x_1 in the transmission range is awake and

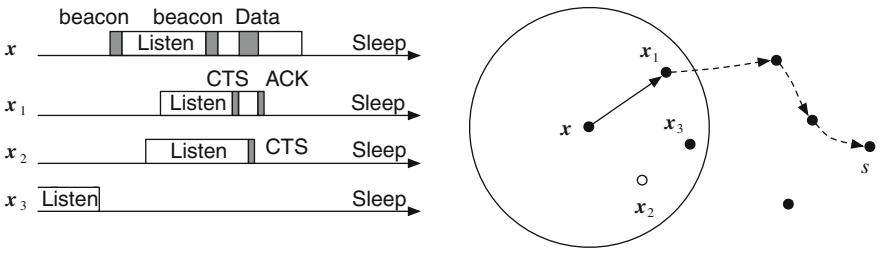


Fig. 1 The transmission process and routing path for a packet with the anycast protocol

hears the packet, it checks for the following criteria: (1) node x_1 is closer to the sink than x, (2) the signal-to-noise ratio (SNR) of the received RTS packet, ψ, is greater than some predefined threshold ψ_{th}, and (3) if the node does not have any packets to send. If all criteria are met, node x_1 sends a Clear-to-Send (CTS) packet. The same conditions may be satisfied at other nodes (e.g., node x_2), which send their CTS as well. Then, node x chooses the first node that sends a CTS packet as the next-hop node and transmits the data packet to it. Successful data packet transmissions are acknowledged by the receiver, otherwise the sender retransmits the data packet until successful.

3 Single-Hop and End-to-End Delay Distribution

As WSNs are increasingly used in critical applications, delay becomes one of the most important metrics in their design. Complex and cross-layer interactions in multi-hop WSNs require a complete stochastic characterization of the delay. Because of the randomness in wireless communication and the low power nature of the communication links in WSNs, average delay or worst-case delay provides limited insight into the operations of WSNs. Probabilistic analysis of delay has been performed for broadcast networks [6, 51, 58, 59, 61] considering several medium access control (MAC) protocols. Indeed, the cumulative distribution function (*cdf*) of the delay for a given deadline can be used as a probabilistic metric for reliability and timeliness. In addition to single-hop delay, additional delay due to multi-hop communication, queuing delay, and wireless channel errors have to be captured as they are imperative to completely characterize the delay distribution in WSNs.

In this section, we discuss the stochastic analysis of delay in WSNs. The resulting delay distribution is an important metric to evaluate the communication services provided by the network, since it measures the probability that the network meets a given deadline. The developed framework highlights the *relationship between network parameters and the delay distribution* in multi-hop WSNs. Using this framework, real-time scheduling, deployment, admission control, and communication solutions can be developed to provide probabilistic QoS guarantees.

3.1 Background

Historically, the problem of probabilistic end-to-end delay has attracted a large interest. The concept of Network Calculus [15] can be extended to support probabilistic delay *bounds* in [9, 22, 34, 60]. Network calculus and its probabilistic extensions are based on a min-plus algebra to provide traffic curves and service curves, which are deterministic (or statistical) bounds of traffic rate and service time, respectively. Accordingly, the *worst case* performance bounds can be analyzed. This approach has limited applications in WSNs due to the randomness in wireless communication,

large variance in the end-to-end delay, the fact that most applications tolerate packet loss for a lower delay of higher priority packets since the efficiency of the system is improved. Accordingly, *probabilistic delay characteristics* are of interest in addition to the worst case bounds.

Real-time theory and queueing theory are combined to provide stochastic models for unreliable networks [38, 75] for networks with heavy traffic rates. The approach in this section is similar to the real-time queueing theory [38] in that a stochastic queuing model for the analysis is used. However, we do not focus on the real-time scheduling problem [38, 40, 75] but instead characterize delay for the development of communication solutions.

Recently, the delay distribution of medium access control (MAC) protocols has been analyzed for IEEE 802.11b DCF protocol [6, 59, 61], IEEE 802.15.4 protocol [56, 58], and TDMA protocols [51], where a broadcast network is considered. Saturated traffic cases are also investigated extensively [6, 59, 61]. In addition, distribution of link layer retransmissions [31], end-to-end delay in a linear network with infitie queues [72], and delay in data aggregation networks [25] are recently modeled. These theoretical approaches are also complemented with empirical approaches that use on-the-fly measurements estimate probabilistic characteristics of end-to-end delay [21, 26, 54].

The remainder of this section is an extension to this state-of-the art to capture multi-hop communication effects, hidden node problems, and the low traffic rate of WSNs. In the following, we first provide a problem definition and then, discuss the probabilistic end-to-end delay analysis model.

3.2 Problem Definition and System Model

Let us assume a network deployment, where each node is indexed by its location $\boldsymbol{x}$. We are interested in the following two problems:

(1) What is the probability distribution function (*pdf*) of single-hop delay, $f_{\mathrm{sh}(\boldsymbol{x},\boldsymbol{y})}(t)$, between two nodes $\boldsymbol{x}$ and $\boldsymbol{y}$ for a new arriving packet?
(2) Given the single-hop delay distribution, what is the *pdf* of the end-to-end delay, $f_{\mathrm{e2e}(\boldsymbol{x},\boldsymbol{s})}(t)$, between a node $\boldsymbol{x}$ and a sink located at $\boldsymbol{s}$?

In the following, we provide an overview of solutions for the two problems above and the detailed descriptions are deferred to Sects. 3.3 and 3.4.

3.2.1 Single-Hop Delay Distribution

In this setting, we can model each node according to a queuing model, which is characterized by its inter-arrival distribution and service process. More specifically, the traffic inter-arrival can be modeled according to a Geometric distribution [68].

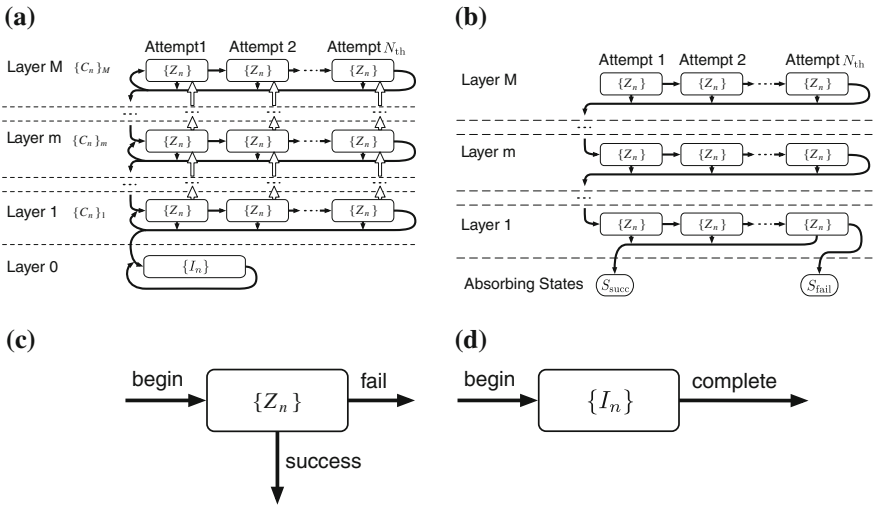


Fig. 2 The structures of Markov chains: **a** $\{X_n\}$ and **b** $\{Y_n\}$. Their building blocks are also shown: **c** $\{Z_n\}$ and **d** $\{I_n\}$

The Geometric distribution is selected as experiment results with different types of traffic (event-based and periodic) indicate that it is a valid model for WSN traffic [68]. Furthermore, a Discrete Time Markov Process (DTMP) is used to model the service behavior. In such a model, the service time is Phase-Type (PH) distributed by definition [52]. Considering a single processor at each node and a queue capacity of M, the resulting model is a discrete time Geom/PH/1/M queueing model.

The communication system at each node is modeled as a discrete-time recurrent Markov chain, $\{X_n\}$. As shown in Fig. 2a, this discrete time Markov chain (DTMC) has a layered structure. Each layer i contains the part of the chain where there are i packets in the queue. The communication behaviors of each node are represented by transitions among states in $\{X_n\}$. Then, a second DTMC, $\{Y_n\}$, which is the absorbing variant of $\{X_n\}$, is used to obtain the single-hop delay distribution. The steady-state parameters found in $\{X_n\}$ are used in $\{Y_n\}$ to derive the associated distributions. The detailed explanation of these DTMCs is provided in Sect. 3.3.

3.2.2 End-to-End Delay Distribution

With each hop modeled as a Geom/PH/1/M queue, the entire network can be considered as a queueing network. Nodes are interrelated according to the traffic constraints. More specifically, the successfully transmitted traffic rate from one node should equal the sum of the incoming relay traffic rate at each of the next-hop neighbors of the node.

The topology of the queueing network depends on the routing protocol used. In this discussion, we focus on the class of routing protocols wherein each node maintains

a probabilistic routing table for its neighbors, e.g., anycast routing protocols [2]. By first calculating the relay traffic and the single hop delay distribution for each pair of nodes, the end-to-end delay can be obtained using an iterative procedure as explained in Sect. 3.4.

3.3 Single-Hop Delay Distribution

For single-hop delay distribution, we are interested in capturing the delay in transmitting a packet from one node to another. This delay includes the queueing delay at the transmitting node and the communication delay due to transmission time, medium access, and wireless channel errors and retransmissions. The communication system at each node can be modeled by a DTMC $\{X_n\}$ and its absorbing variant $\{Y_n\}$. The first DTMC, $\{X_n\}$, captures the equilibrium behavior of the system. The second DTMC, $\{Y_n\}$, is then used to analyze the transient communication behavior after a specific packet arrives. The single-hop delay of the packet transmission is then represented as the absorption time of $\{Y_n\}$. In the following, we will discuss the construction of $\{X_n\}$ and $\{Y_n\}$ and show how the single-hop delay distribution is derived.

3.3.1 Steady-State Analysis

The DTMC, $\{X_n\}$, as shown in Fig. 2a, is two dimensional, where the horizontal dimension captures the transmission attempts and the vertical dimension captures the queue state. Accordingly, $\{X_n\}$ is composed of $M + 1$ layers, where each layer m $(0 \leq m \leq M)$ represents the number of packets in the queue and M is the queue capacity. The first layer, the quiescent layer, $\{I_n\}$, $(m = 0)$ represents the *quiescent process*, during which the node does not have any packet to send, and waits for new packets. The communication layers, $\{C_n\}_m$ $(\mathrm{m} > 0)$, represent the *communication process* in which packets are transmitted. One or more transmission attempts are conducted, until either the packet is successfully transmitted, or the maximum number of transmission attempts, N_{tx}, is exceeded. In the latter case, the packet is dropped.

A layer m in $\{X_n\}$ is denoted by $\{C_n\}_m$, and is composed of N_{tx} blocks. The bth block in layer m is denoted by $\{Z_n\}_{m,b}$.[1] As shown in Fig. 2c, each block models a single transmission attempt. The structure of $\{Z_n\}$ depends on the MAC protocol used. Packets are dropped if they arrive at a full queue or if all N_{tx} transmission attempts fail. Consequently, the vth state in layer m and transmission attempt b is denoted by $S_{m,b,v}$.

In a typical network operation, each node is exposed to two types of incoming traffic: locally generated and relay traffic. While the locally generated traffic can arrive at any time and depends on the application, the relay traffic can only arrive

[1] In the following, we drop the indices m and b, where appropriate, to simplify the notation.

when the node is listening. Therefore, the total traffic rate depends on the state of the process. Let us denote the locally generated traffic rate and the relay traffic rate for a node by λ_{lc} and λ_{re}, respectively. Then, in the states where the node is listening, the total traffic rate is $\lambda_{\mathrm{lc}} + \lambda_{\mathrm{re}}$. When the radio is in sleep, the total traffic rate is only λ_{lc}.

The quiescent and the communication layers, $\{I_n\}$ and $\{C_n\}$, are parameterized by the following notations:

- $\boldsymbol{P}_I$ and $\boldsymbol{P}_C$: the transition probability matrices among the states in $\{I_n\}$ and $\{C_n\}$, respectively.
- $\boldsymbol{\alpha}_I$ and $\boldsymbol{\alpha}_C$: the initial probability vector for $\{I_n\}$ and $\{C_n\}$, respectively.
- $\boldsymbol{t}_I^{\mathrm{s}}$ and $\boldsymbol{t}_C^{\mathrm{s}}$: the probability vector from each state in $\{I_n\}$ and $\{C_n\}$ to complete the quiescent process and the communication process successfully, respectively.
- $\boldsymbol{t}_C^{\mathrm{f}}$: the probability vector from each state in $\{C_n\}$ to complete the communication process unsuccessfully.
- $\boldsymbol{\lambda}_I$ and $\boldsymbol{\lambda}_C$: the packet arrival probability vector for each state in $\{I_n\}$ and $\{C_n\}$, respectively. Each element in the vectors equals to the probability of a new packet arrival in a time unit when the process is in the corresponding state.

The Markov chain block for each transmission attempt, $\{Z_n\}$, is characterized by the following:

- $\boldsymbol{P}_Z$, the transition probability matrix among the states in $\{Z_n\}$,
- $\boldsymbol{\alpha}_Z$, the initial probability vector for $\{Z_n\}$, and
- $\boldsymbol{t}_Z^{\mathrm{s}}$ and $\boldsymbol{t}_Z^{\mathrm{f}}$, the probability vector from each state in $\{Z_n\}$ to complete the transmission attempt successfully or unsuccessfully, respectively.

The states and the transitions related to $\{Z_n\}$ depend on the MAC protocol employed. For now, let us assume that these matrices are known and leave the discussion on how to obtain them to the case study in Sect. 3.5. Then, the transition probability matrix among the states in a single layer $\{C_n\}$ in $\{X_n\}$ is

$$\boldsymbol{P}_C = \begin{bmatrix} \boldsymbol{P}_Z & \boldsymbol{t}_Z^{\mathrm{f}}\boldsymbol{\alpha}_Z & & \boldsymbol{0} \\ & \ddots & \ddots & \\ & & \boldsymbol{P}_Z & \boldsymbol{t}_Z^{\mathrm{f}}\boldsymbol{\alpha}_Z \\ \boldsymbol{0} & & & \boldsymbol{P}_Z \end{bmatrix}, \tag{1}$$

where the number of $\boldsymbol{P}_Z$ blocks in $\boldsymbol{P}_C$ is equal to N_{tx}, i.e., the maximum number of attempts for each packet transmission. Similarly, the initial probability vector, $\boldsymbol{\alpha}_C$, and the probability vectors, $\boldsymbol{t}_C^{\mathrm{s}}$ and $\boldsymbol{t}_C^{\mathrm{f}}$ to complete a layer in success and failure are

$$\boldsymbol{\alpha}_C = \begin{bmatrix} \boldsymbol{\alpha}_Z & \boldsymbol{0} & \cdots & \boldsymbol{0} \end{bmatrix} \tag{2}$$

$$\boldsymbol{t}_C^{\mathrm{s}} = \begin{bmatrix} \boldsymbol{t}_Z^{\mathrm{s}} & \boldsymbol{t}_Z^{\mathrm{s}} & \cdots & \boldsymbol{t}_Z^{\mathrm{s}} \end{bmatrix}^{\mathrm{T}} \tag{3}$$

$$\boldsymbol{t}_C^{\mathrm{f}} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{t}_Z^{\mathrm{f}} \end{bmatrix}^{\mathrm{T}} \tag{4}$$

respectively.

The transition probability matrix, $\boldsymbol{Q}_X$, of the entire Markov chain $\{X_n\}$ can then be found according to transitions between different states at each layer [68, 71] as follows:

$$\boldsymbol{Q}_X = \begin{array}{c} \text{layer} \\ 0 \\ 1 \\ 2 \\ \cdots \\ M \end{array} \begin{array}{c} \begin{array}{ccccc} 0 & 1 & 2 & \cdots & M \end{array} \\ \begin{pmatrix} \boldsymbol{A}_{\text{s0}} & \boldsymbol{A}_{\text{u0}} & & & \boldsymbol{0} \\ \boldsymbol{A}_{\text{d0}} & \boldsymbol{A}_{\text{s}} & \boldsymbol{A}_{\text{u}} & & \\ & \boldsymbol{A}_{\text{d}} & \ddots & \ddots & \\ & & \ddots & \boldsymbol{A}_{\text{s}} & \boldsymbol{A}_{\text{u}} \\ \boldsymbol{0} & & & \boldsymbol{A}_{\text{d}} & \boldsymbol{A}_{\text{s}} + \boldsymbol{A}_{\text{u}} \end{pmatrix} \end{array}, \tag{5}$$

where each nonzero block corresponds to the transition probability among all layers. The duration of the time unit T_{u} is chosen to be small enough such that the probability of having two or more transitions in a single time unit is negligible. Therefore, it is only possible for $\{X_n\}$ to have intra-layer transitions and inter-layer transitions to adjacent layers.

The first row and column of blocks in $\boldsymbol{Q}_X$ corresponds to the transition probabilities from and to the quiescent layer, respectively. Then, the equilibrium state probability vector, $\boldsymbol{\pi}$, for $\{X_n\}$ is calculated by solving $\boldsymbol{\pi}\boldsymbol{Q}_X = \boldsymbol{\pi}$ and $\sum_i \boldsymbol{\pi}_i = 1$. We will illustrate this process with a basic example next.

Example 1. Let us now consider an example protocol, where a node conducts a duty cycle operation every 2 s. It first sleeps for 1 s and then listens to the channel for another 1 s. If a packet is received during the listening period with a probability λ_{re}, or if a local packet is generated in any period with a probability of λ_{lc}, the node attempts to transmit the packet. The transmission attempt takes 1 s with a failure rate p, and the node persistently attempts to transmit the packet until successful. While transmitting, the node cannot receive any packets, but can still generate packets. The queue length is $M = 2$.

For this protocol, a time unit of 1 s can be chosen since all time periods are 1 s. Then, the quiescent process can be modeled by two states, and the communication process can be modeled by one state, as shown in Fig. 3. The quiescent process,

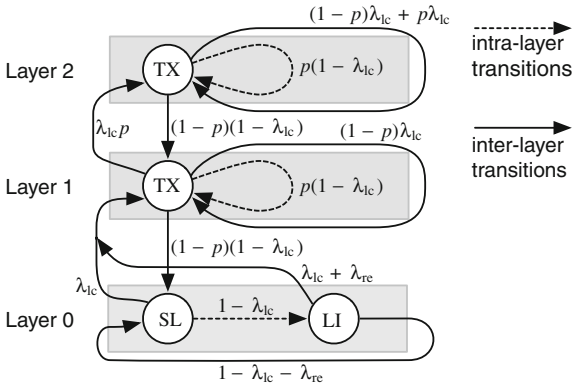


Fig. 3 The structure of $\{X_n\}$ for the example

$\{I_n\}$, contains a sleeping state (SL) and a listening state (LI), whereas the communication process, $\{C_n\}$, contains a single transmission state (TX). Accordingly, $\boldsymbol{P}_I, \boldsymbol{P}_C, \boldsymbol{\alpha}_I, \boldsymbol{\alpha}_C, \boldsymbol{t}_I^{\mathrm{s}}, \boldsymbol{t}_C^{\mathrm{s}}, \boldsymbol{t}_C^{\mathrm{f}}, \boldsymbol{\lambda}_I$ and $\boldsymbol{\lambda}_C$ are found as:

$$\begin{aligned}
&\boldsymbol{P}_I = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, && \boldsymbol{P}_C = p, \\
&\boldsymbol{\alpha}_I = \begin{bmatrix} 1 & 0 \end{bmatrix}, && \boldsymbol{\alpha}_C = 1, \\
&\boldsymbol{t}_I^{\mathrm{s}} = \begin{bmatrix} 0 & 1 \end{bmatrix}^{\mathrm{T}}, && \boldsymbol{t}_C^{\mathrm{s}} = 1 - p, \qquad \boldsymbol{t}_C^{\mathrm{f}} = 0, \\
&\boldsymbol{\lambda}_I = \begin{bmatrix} \lambda_{\mathrm{lc}} & \lambda_{\mathrm{lc}} + \lambda_{\mathrm{re}} \end{bmatrix}, && \boldsymbol{\lambda}_C = \lambda_{\mathrm{lc}},
\end{aligned} \tag{6}$$

where $\boldsymbol{t}_C^{\mathrm{f}} = 0$ because the communication persistently attempts to transmit until successful, thus it can never fail. Therefore, the blocks in $\boldsymbol{Q}_X$ (see (5)) are expressed as

$$\begin{aligned}
&\boldsymbol{A}_{\mathrm{u}} = \lambda_{\mathrm{lc}} p, && \boldsymbol{A}_{\mathrm{u0}} = [\lambda_{\mathrm{lc}} \;\; \lambda_{\mathrm{lc}} + \lambda_{\mathrm{re}}]^{\mathrm{T}} \\
&\boldsymbol{A}_{\mathrm{s}} = \lambda_{\mathrm{lc}}(1-p) + (1-\lambda_{\mathrm{lc}})p, && \boldsymbol{A}_{\mathrm{s0}} = \begin{bmatrix} 0 & 1-\lambda_{\mathrm{lc}} \\ 1-\lambda_{\mathrm{lc}}-\lambda_{\mathrm{re}} & 0 \end{bmatrix} \\
&\boldsymbol{A}_{\mathrm{d}} = (1-\lambda_{\mathrm{lc}})(1-p), && \boldsymbol{A}_{\mathrm{d0}} = [(1-\lambda_{\mathrm{lc}})(1-p) \;\; 0]
\end{aligned} \tag{7}$$

3.3.2 Transient Analysis

Given the steady state characteristics of the system, we can now find the distribution of single-hop delay for a packet by an absorbing DTMC. Consider a particular packet that enters the system at time $t = t_0$. The single-hop delay of the packet is the time spent until it is transmitted or dropped in the system. To derive the delay distribution, we use an absorbing variant of $\{X_n\}$ that is denoted as $\{Y_n\}$ an shown in Fig. 2b. In $\{Y_n\}$, the quiescent layer of $\{X_n\}$ is replaced by two absorbing states S_{succ} and S_{fail}, corresponding to the two cases where the packet is successfully transmitted and dropped, respectively. In addition, all new packet arrivals are ignored since they do not interfere with the service time of the packet concerned in a first in first out (FIFO) queue. Thus, the state transitions occur only inside a layer or from layer $m+1$ to m. The steps to obtain $\{Y_n\}$ from $\{X_n\}$ is as follows.

Before the packet arrives, the system is in one of the states according to the equilibrium state probability vector, $\boldsymbol{\pi}$. After the new packet arrives, if the queue is full, the packet is immediately dropped. The probability of queue full is

$$p_{\mathrm{qf}} = \boldsymbol{\pi}_M \boldsymbol{A}_{\mathrm{u}} \mathbf{1}, \tag{8}$$

where $\boldsymbol{\pi}_M$ is the sub-vector in $\boldsymbol{\pi}$ corresponding to the Mth layer. Otherwise, the packet is inserted into the queue. The probability vector that the node is in a specific state after the new packet arrives is $\boldsymbol{\pi}' = \boldsymbol{\pi}\, \boldsymbol{Q}_Y^{\mathrm{up}}$, where $\boldsymbol{Q}_Y^{\mathrm{up}}$ is the transition probabil-

ity matrix of $\{Y_n\}$ conditioned on the fact that the new packet arrives. $\boldsymbol{Q}_Y^{\text{up}}$ is derived from $\boldsymbol{Q}_X$ in (5) by replacing $\boldsymbol{\lambda}_I$ and $\boldsymbol{\lambda}_C$ with vectors of all 1's and replacing $\boldsymbol{A}_{\text{s}} + \boldsymbol{A}_{\text{u}}$ with $\boldsymbol{A}_{\text{s}}$. Note that $\boldsymbol{A}_{\text{u}}$ in the bottom-right block accounts for the transition that will cause a packet to drop because of a full queue. Then, $\boldsymbol{\pi}'$ is the initial probability vector for $\{Y_n\}$.

Accordingly, the transition probability matrix for $\{Y_n\}$ is

$$\boldsymbol{Q}_Y = \begin{bmatrix} 1 & 0 & \mathbf{0} \\ 0 & 1 & \mathbf{0} \\ \boldsymbol{t}_Y^{\text{s}} & \boldsymbol{t}_Y^{\text{f}} & \boldsymbol{P}_Y \end{bmatrix}, \tag{9}$$

where the transition probabilities from and to the absorbing states S_{succ} and S_{fail} are listed in the first two rows and columns, respectively. The transition probability matrix among the transient states, i.e., all states except S_{succ} and S_{fail}, is given by

$$\boldsymbol{P}_Y = \begin{bmatrix} \boldsymbol{P}_C & & & \mathbf{0} \\ \boldsymbol{t}_C\boldsymbol{\alpha}_C & \boldsymbol{P}_C & & \\ & \ddots & \ddots & \\ \mathbf{0} & & \boldsymbol{t}_C\boldsymbol{\alpha}_C & \boldsymbol{P}_C \end{bmatrix}. \tag{10}$$

This is obtained from (5) by removing the first row and first column of blocks, and replacing $\boldsymbol{\lambda}_I$ and $\boldsymbol{\lambda}_C$ with vectors of all 0's for each remaining block. The transition probability vectors from each of the transient states to the absorbing states are

$$\boldsymbol{t}_Y^{\text{s}} = \left[\boldsymbol{t}_C^{\text{s}}\ \mathbf{0}\ \mathbf{0}\ \cdots\right]^{\text{T}}, \boldsymbol{t}_Y^{\text{f}} = \left[\boldsymbol{t}_C^{\text{f}}\ \mathbf{0}\ \mathbf{0}\ \cdots\right]^{\text{T}}, \tag{11}$$

respectively, where $\boldsymbol{t}_C^{\text{s}}$ and $\boldsymbol{t}_C^{\text{f}}$ are given in (3) and (4), respectively. Finally, since a transition in $\{Y_n\}$ takes a time unit T_{u}, the following important results are directly obtained:

Theorem 1. *The probability mass function (*pmf*) of the number of time units, K, a packet should wait before being transmitted and dropped are*

$$f_K^s(k) = \boldsymbol{\alpha}_Y \boldsymbol{P}_Y^{k-1} \boldsymbol{t}_Y^s,\ f_K^f(k) = \boldsymbol{\alpha}_Y \boldsymbol{P}_Y^{k-1} \boldsymbol{t}_Y^f, \tag{12}$$

respectively, where $\boldsymbol{\alpha}_Y = (\boldsymbol{\pi}'_1, \boldsymbol{\pi}'_2, \cdots, \boldsymbol{\pi}'_M)$, i.e., $\boldsymbol{\pi}'$ without the elements corresponding to the quiescent layer, and $\boldsymbol{P}_Y^{k-1}$ represents the $(k-1)$th power of $\boldsymbol{P}_Y$.

Proof. The theorem follows from [50, Ch. 9.5].

The *pmf* of the number of time units a packet should wait, regardless of being transmitted and dropped, is obtained by adding $f_K^{\text{s}}(k)$ and $f_K^{\text{f}}(k)$. Thus, the following corollary is directly obtained.

Corollary 1. *The* pmf *of single-hop delay, measured by the number of time units of T_u, is given by*

$$f_K(k) = \boldsymbol{\alpha}_Y \boldsymbol{P}_Y^{k-1} \boldsymbol{t}_Y. \tag{13}$$

Using this model, the probability that the packet is eventually delivered in success can also be found, and is given by the following corollary:

Corollary 2. *The delivery rate of a new arriving packet is*

$$p_{\text{deli}} = \sum_{k=1}^{+\infty} f_K^s(k) = \boldsymbol{\alpha}_Y (\boldsymbol{I} - \boldsymbol{P}_Y)^{-1} \boldsymbol{t}_Y^s. \tag{14}$$

Accordingly, the first two moments of the single-hop delay can also be derived. These moments are widely used as the performance metrics in WSN applications.

Corollary 3. *The mean and variance of single-hop delay for a new arriving packet are*

$$\mu_K = \frac{\boldsymbol{\alpha}_Y (\boldsymbol{I} - \boldsymbol{P}_Y)^{-2} \boldsymbol{t}_Y^s}{p_{\text{deli}}}, \tag{15}$$

$$\sigma_K^2 = \frac{\boldsymbol{\alpha}_Y (2(\boldsymbol{I} - \boldsymbol{P}_Y)^{-3} - (\boldsymbol{I} - \boldsymbol{P}_Y)^{-2}) \boldsymbol{t}_Y^s}{p_{\text{deli}}} - \mu_K^2, \tag{16}$$

respectively.

The derivations are straightforward since

$$\mu_K = \mathrm{E}[K] = \frac{\sum_{k=1}^{+\infty} k \cdot f_K^{\mathrm{s}}(k)}{\sum_{k=1}^{+\infty} f_K^{\mathrm{s}}(k)}, \tag{17}$$

$$\sigma_K^2 = \mathrm{E}[K^2] - (\mathrm{E}[K])^2 = \frac{\sum_{k=1}^{+\infty} k^2 \cdot f_K^{\mathrm{s}}(k)}{\sum_{k=1}^{+\infty} f_K^{\mathrm{s}}(k)} - \mu_K^2, \tag{18}$$

where E[] represents expectation.

Based on the single-hop delay distribution, we will next derive the end-to-end delay distribution in WSNs.

3.4 End-to-End Delay Distribution

In a multi-hop WSN, the delay is generally calculated as the time it takes for a packet generated in the network to reach the sink. This delay, accordingly, depends on the route taken by the packet. Thus, the topology of the network and the routing protocol defines the end-to-end delay performance. Next, we discuss the case of random deployments.

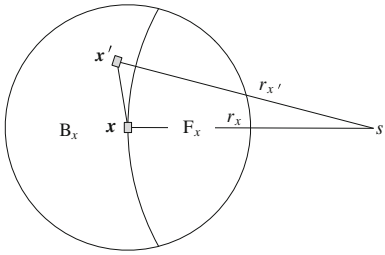


Fig. 4 The feasible region, $\mathbb{F}_{\boldsymbol{x}}$, and the infeasible region, $\mathbb{B}_{\boldsymbol{x}}$, of node $\boldsymbol{x}$

For the random deployment, the nodes are assumed to be located in the network according to a Poisson point process with density ρ. For these types of networks, geographic routing protocols [2] are often used due to their scalability and adaptability to the random geographic locations of the nodes. In such protocols, instead of the routing probability $p^{\text{fw}}_{i,j}$ between any pair of nodes i and j, the routing probability between any pair of *locations* $\boldsymbol{x}$ and $\boldsymbol{y}$, $p^{\text{fw}}_{\boldsymbol{x},\boldsymbol{y}}$ is of interest.

A common scenario is considered, where the nodes in the network generate the same amount of local traffic to a sink. Moreover, each node $\boldsymbol{x}$ forwards packets to the neighboring nodes within its *feasible region* $\mathbb{F}_{\boldsymbol{x}}$, i.e., the region in which nodes are closer to the sink, but are still in the transmission range. Assume that the sink is located at the center of a circular plane with a radius R. In this scenario, the end-to-end delay analysis can take advantage of the symmetry of the topology as explained next.

The entire circular plane is discretized into concentric rings indexed by their distance to the sink, r. Each node senses the physical events, and generates packets with traffic rate λ_{lc}. By symmetry, the relay traffic $\lambda_{\text{re},r}$ is the same for all nodes in the same ring r. We assume a polar coordinates system with the sink located at the origin.

As shown in Fig. 4, for a node $\boldsymbol{x}$ located at $\boldsymbol{x} = (r_{\boldsymbol{x}}, \theta_{\boldsymbol{x}})$, the relay traffic arrives from any node $\boldsymbol{y}$ in the *infeasible region* $\mathbb{B}_{\boldsymbol{x}} = \mathbb{C}_{\boldsymbol{x}} \setminus \mathbb{F}_{\boldsymbol{x}}$, i.e., the region in which nodes are farther to the sink but are still in the transmission range. To derive the relay traffic rate for $\boldsymbol{x}$ and other nodes in ring $r_{\boldsymbol{x}}$, consider the small area $(r_{\boldsymbol{x}} : r_{\boldsymbol{x}} + \Delta r, \theta : \theta + \Delta\theta)$ around node $\boldsymbol{x}$ located at $(r_{\boldsymbol{x}}, \theta)$. Similar to the deterministic deployment, the relay traffic rate $\lambda_{\text{re},r_{\boldsymbol{x}}}$ is given by

$$\begin{aligned} \lambda_{\text{re},r_{\boldsymbol{x}}} &= \bar{\lambda}_{\text{re},r_{\boldsymbol{x}}} / \pi^{\text{listen}}_{r_{\boldsymbol{x}}}, \\ \bar{\lambda}_{\text{re},r_{\boldsymbol{x}}} &= \frac{\int_{\mathbb{B}_{\boldsymbol{x}}} \rho(\bar{\lambda}^{\text{f}}_{\boldsymbol{y}} + \lambda_{\text{lc}}) p^{\text{fw}}_{\boldsymbol{y},\boldsymbol{x}} \; p_{\text{deli},\boldsymbol{y},\boldsymbol{x}} \, \mathrm{d}\boldsymbol{y}}{\rho \Delta r \Delta\theta r_{\boldsymbol{x}}}, \end{aligned} \tag{19}$$

where ρ is the network density of the Poisson node distribution, $p^{\text{fw}}_{\boldsymbol{y},\boldsymbol{x}}$ and $p_{\text{deli},\boldsymbol{y},\boldsymbol{x}}$ are similarly defined as $p^{\text{fw}}_{m,i}$ and $p_{\text{deli},m,i}$, except that the nodes are indexed by their locations.

Finally, $p^{\text{fw}}_{\boldsymbol{y},\boldsymbol{x}}$ in (19) is the routing protocol-specific probability that the node at $\boldsymbol{y}$ transmit packets to a node at $\boldsymbol{x}$. A case study for the anycast protocol will be provided in Sect. 3.5 to show how this probability is obtained.

Thus, according to (19), the traffic rate of node $\boldsymbol{x}$ at each state can be determined. Accordingly, the input traffic rate vectors $\boldsymbol{\lambda}_I$ and $\boldsymbol{\lambda}_C$ of node $\boldsymbol{x}$ can be found according to Sect. 3.3. Then, the equilibrium state probability for the DTMC $\{X_n\}$, $\boldsymbol{\pi}_{r_{\boldsymbol{x}}}$ is obtained. Note that in (19), the traffic rate for nodes in ring $r_{\boldsymbol{x}}$ depends on the traffic rate and delivery rate for nodes in their infeasible region. Therefore, the single-hop delay distribution is obtained first for nodes in the outmost ring, and then the inner rings in the decreasing order of the ring radius.

By symmetry, the end-to-end delay distribution to the sink is the same for all nodes with a same distance $r_{\boldsymbol{x}}$ to the sink, and is obtained by

$$f_{\text{e2e}(r_{\boldsymbol{x}})}(t) = \int_{\mathbb{F}_{\boldsymbol{x}}} p^{\text{fw}}_{\boldsymbol{x},\boldsymbol{y}} f_{\text{sh}(r_{\boldsymbol{x}})} * f_{\text{e2e}(r_{\boldsymbol{y}})}(t)\mathrm{d}\boldsymbol{y}. \tag{20}$$

The end-to-end delay distribution is found in the ascending order of the distance to the sink.

Next, the anycast protocol is used as a case study to illustrate the end-to-end delay analysis in a randomly deployed network.

3.5 Case Study: Anycast Protocol

Most WSN communication protocols employ a duty cycle mechanism to save energy. Here, we consider an anycast protocol, which aims to find next hops on the fly to forward packets to the sink. Accordingly, we show how the single-hop and the end-to-end delay analysis in Sects. 3.3 and 3.4 can be applied to protocols with *duty cycle* operations for a randomly deployed network.

We first show the DTMC $\{X_n\}$ for the protocol. Then, the protocol-specific parameters for the generic analysis in Sect. 3.3, including the relay traffic rate at each state, and the transition probabilities for $\{X_n\}$ are derived. The single-hop delay distribution for each pair of nodes is obtained after these parameters are known. Finally, the end-to-end delay distribution from each node to the sink is provided.

3.5.1 Markov Process Overview

The structures of $\{I_n\}$ and $\{C_n\}$ in DTMC $\{X_n\}$ for this protocol are shown in Fig. 5. The quiescent layer $\{I_n\}$ consists of a chain of sleeping states and a chain of listening states of duration T_{u}. One may think that a single sleeping state and a single listening state are enough to model the duty cycle operation, similar to the example. However, because of the memoryless nature of Markov process, arbitrary values of duty cycle must be captured with a specific number of states representing the active period and sleeping period.

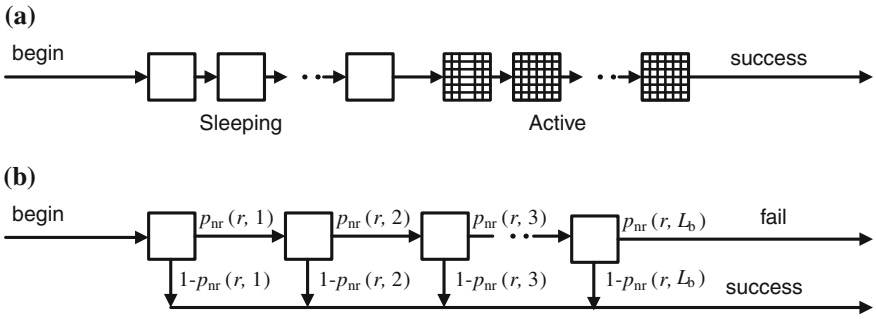


Fig. 5 The Markov chain structure of (**a**) the quiescent process, $\{I_n\}$, and (**b**) the communication process, $\{C_n\}$, for the anycast protocol

When there is no communication, the Markov process transitions through sleeping states and listening states periodically, representing the duty cycle operation. In the listening states, the node listens to the channel. Thus, both locally generated packets and relay packets can arrive. In the sleeping states, the node turns off its transceiver and only local packets can arrive. The number of states in $\{I_n\}$ is $L_c = T_{sl}/T_u + T_a/T_u = T_p/T_u$, where T_u is the time unit, and T_p is the duration of a duty-cycle period. The values of T_{sl} and T_a depend on the protocol parameters. A large T_u can reduce the number of states in the DTMC, thus, reducing complexity of the model, but at the cost of reducing the granularity and accuracy of the result.

When a packet arrives, the node terminates the quiescent process and begins the first layer of communication process $\{C_n\}$. In each $\{C_n\}$ layer, the node keeps transmitting beacon packets. The number of states in $\{C_n\}$ is $L_b = T_b/T_u$, where T_b is the beacon time-out.

If a node receives RTS responses from other nodes, it starts transmitting the data packet to the first responding node. Retransmissions are conducted in case of a transmission failure. Since only neighbor nodes that receive the beacon packets with a high SNR will response, a high quality wireless channel is guaranteed. Moreover, in most WSN applications, the traffic rate is low, and the chance of packet collision with other nodes is small. Therefore, data packets transmitted successfully in limited number of (re)transmission attempts, which takes negligible time compared to the sleeping cycle T_p (usually longer than 10 s). Thus, $\{C_n\}$ only contains transmission states. In cases where retransmissions take longer durations, $\{C_n\}$ can be extended with additional states. When the first RTS packet is received, the transmission terminates in a success. When the beacon transmission times out, the packet is dropped, and the transmission terminates in a failure. In either way, the node enters the lower layer. Note that the beacon timeout T_b is usually chosen equal to the cycle T_p. This is to ensure that each neighbor node can receive the beacon messages within their duty cycle period. The entire beacon communication process before packet delivery or timeout is regarded as a single transmission attempt. Thus, each communication layer $\{C_n\}$ contains only one block of $\{Z_n\}$.

3.5.2 Steady State and Transient Analyses

The transition probability matrices in $\{I_n\}$ and $\{C_n\}$, are obtained according to the Markov structure in Fig. 5. In either $\{I_n\}$ or $\{C_n\}$, there is only one initial state (denoted by "begin") with probability of 1. States with outgoing transitions denoted by "success" or "fail" have a probability to complete the current process in a success or failure, respectively. The transition probabilities among states are shown in Fig. 5. Note that transitions with a probability of 1 are not labeled. The transition probabilities $p_{\mathrm{nr}}(r, v)$, $(1 \le v \le L_{\mathrm{b}})$, and the traffic rate $\boldsymbol{\lambda}_I, \boldsymbol{\lambda}_C$ are explained in the following.

In the jth time unit in $\{C_n\}$, a node located at $\boldsymbol{x}$ in ring r has a probability of $p_{\mathrm{nr}}(r, v)$ of not receiving any CTS response, and enters the next state. If in all L_{b} states, the node receives no CTS response, the transmission fails and the packet is dropped. On the other hand, if in any of the states, a CTS response is received, the node transmits the packet and the transmission succeeds. The probability $p_{\mathrm{nr}}(r, v)$ is the conditional probability that given the transmissions in the previous $v-1$ states fails, the transmissions in the v-th state still fails. For simplicity, the hidden terminals are ignored. Therefore,

$$\begin{aligned} p_{\mathrm{nr}}(r, 1) &= p_{\mathrm{nr}}(r, 1 \sim 1) \\ p_{\mathrm{nr}}(r, v) &= p_{\mathrm{nr}}(r, 1 \sim v) / p_{\mathrm{nr}}(r, 1 \sim v-1), \ 2 \le v \le L_{\mathrm{b}} \end{aligned} \tag{21}$$

where $p_{\mathrm{nr}}(r, 1 \sim v)$ is the probability that during all first v states in $\{C_n\}$, beacon transmission fails, since no CTS packet is received in these states. Accordingly,

$$p_{\mathrm{nr}}(r, 1 \sim v) = \prod_{\boldsymbol{y}=(r_y, \theta_y) \in \mathbb{F}(\boldsymbol{x})} \left(1 - p_{\mathrm{ex}}(r_y)\, p_{\mathrm{ol}}(r_y, v)\, p_{\mathrm{SNR}}(\boldsymbol{x}, \boldsymbol{y})\right), \tag{22}$$

where each of the small areas at $\boldsymbol{y}$ is located within the transmission range of $\boldsymbol{x}$, $\mathbb{C}(\boldsymbol{x})$, and is closer to the sink than $\boldsymbol{x}$ (this range is called the *feasible region of* $\boldsymbol{x}$, $\mathbb{F}(\boldsymbol{x})$, as shown in Fig. 6a); r_y is the distance from the small area to the sink; $p_{\mathrm{ex}}(r_y)$ is the probability that there exists a node in each area, and is given by

$$p_{\mathrm{ex}}(r) = \rho \Delta r \Delta \theta r, \tag{23}$$

where ρ is the node density. Moreover, $p_{\mathrm{ol}}(r_y, v)$ in (22) is the probability that the active period of a node located r_y away from the sink overlaps with the first j beacon transmission time units of the node at $\boldsymbol{x}$; and $p_{\mathrm{SNR}}(\boldsymbol{x}, \boldsymbol{y})$ is the probability that a packet, transmitted from a node at $\boldsymbol{x}$ to a node at $\boldsymbol{y}$, has an SNR higher than some predefined threshold ψ_{th}. It is obtained by (10) in [76].

Next, let us derive the probability that the active period of a node at $\boldsymbol{y}$ overlaps with the first v beacon transmission time units of a node at $\boldsymbol{x}$, $p_{\mathrm{ol}}(r_y, v)$. If node $\boldsymbol{x}$ receives no response in each of the small areas, at least one of the following statements is true: (1) a node does not exist in the area, (2) at least one node exists but they are sleeping

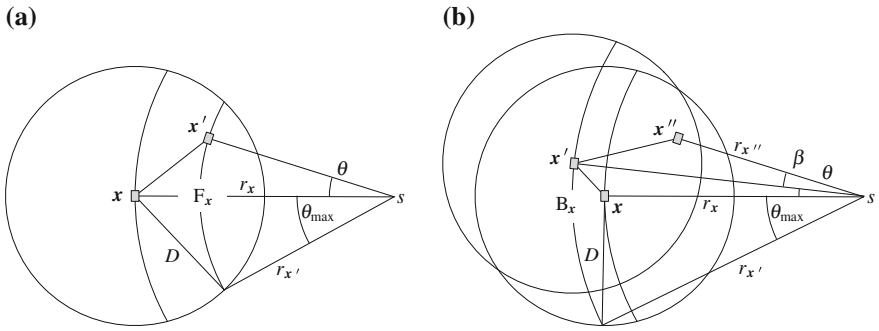


Fig. 6 The feasible region and infeasible region around node $\boldsymbol{x}$, divided into small areas (**a**) Node y is in the feasible region of x (**b**) Node y is in the infeasible region of x, and node z is in the feasible region of y

during any of the first v slots, and (3) at least one node exists and is awake, but the SNR of the beacon packet they receive is lower than the predefined threshold ψ_{th}. Node $\boldsymbol{y}$ is awake during any of the first v slots means that the first beacon transmission time unit of node $\boldsymbol{x}$ either coincides with any of the awake time units of node $\boldsymbol{y}$ or coincides with the last $v-1$ sleeping units of node $\boldsymbol{y}$. Thus, $p_{\text{ol}}(r_y, v)$ is given by

$$p_{\text{ol}}(r_y, v) = \sum_{k=1}^{L_{\text{w}}} \boldsymbol{\pi}_{W_k}(r_y) + \begin{cases} \sum_{k=L_{\text{sl}}-v+1}^{L_{\text{sl}}} \boldsymbol{\pi}_{\text{s}}(r_y), & 1 \le v < L_{\text{sl}} \\ \sum_{k=1}^{L_{\text{sl}}} \boldsymbol{\pi}_{\text{s}}(r_y), & v \ge L_{\text{sl}} \end{cases} \tag{24}$$

where L_{sl} is the number of sleeping time units in $\{I_n\}$, $\boldsymbol{\pi}_{W_k}(r_y)$ and $\boldsymbol{\pi}_{\text{s}}(r_y)$ are the equilibrium probability that node $\boldsymbol{y}$ is in the kth awake state or sleeping state in $\{X_n\}$, respectively. L_{c} and L_{w} are the number of total and awake states in $\{I_n\}$, respectively.

Therefore, $p_{\text{nr}}(r, 1 \sim v)$ in (22) is determined using (23) and (24), and $p_{\text{nr}}(r, v)$ in (21) is obtained using (22).

Now, we focus on the traffic rates at each state, $\boldsymbol{\lambda}_I$ and $\boldsymbol{\lambda}_C$. In sleeping states and listening states, the traffic arrival rate is λ_{lc} and $\lambda_{\text{re}}(r) + \lambda_{\text{lc}}$, respectively. In beacon transmission states, since nodes are assumed not to respond to any relay packets, the traffic rate is λ_{lc}.

If we consider the small area $(r : r + \Delta r, \theta : \theta + \Delta\theta)$, the forwarded traffic arrives from any node $\boldsymbol{y} = (r_y, \theta_y)$ in the *infeasible region* $\mathbb{B}(\boldsymbol{x}) = \mathbb{C}(\boldsymbol{x}) \setminus \mathbb{F}(\boldsymbol{x})$, as shown in Fig. 6b. Therefore $\lambda_{\text{re}}(r)$ is given by

$$\lambda_{\text{re}}(r) = \frac{\sum_{\boldsymbol{y} \in \mathbb{B}(\boldsymbol{x})} p_{\text{ex}}(r_y) \lambda_{\text{o}}(r_y) p_{\text{fw}}(\boldsymbol{y}, \boldsymbol{x})}{p_{\text{ex}}(r) \pi_{\text{li}}(r)}, \tag{25}$$

where $\lambda_o(r_y)$ is the output traffic transmitted from $\boldsymbol{y}$. $\pi_{li}(r)$ is the probability that node $\boldsymbol{x}$ is in any listening state, and is the sum of the probabilities corresponding to all listening states in $\boldsymbol{\pi}(r)$. Moreover, $\lambda_o(r_y)$ is calculated by

$$\lambda_o(r_y) = \boldsymbol{\lambda}(r_y)(\boldsymbol{\pi}(r_y))^{\mathrm{T}}(1 - p_{\mathrm{qfull}}(r_y) - p_{\mathrm{drop}}(r_y)), \tag{26}$$

where $p_{\mathrm{fw}}(\boldsymbol{y}, \boldsymbol{x})$ is the probability that a node $\boldsymbol{y}$ forwards a packet to node $\boldsymbol{x}$, among all possible forward targets, and $\boldsymbol{\lambda}(r_y)$ is the traffic rate vector for all states in $\{X_n\}$ for $\boldsymbol{y}$. The probability that the packet is dropped due to beacon transmission timeout, $p_{\mathrm{drop}}(r_y)$, is easily obtained as $p_{\mathrm{drop}}(r_y) = p_{\mathrm{nr}}(r, 1 \sim L_b)$ (see (22)). The probability that the queue is full when the packet arrives, $p_{\mathrm{qfull}}(r_y)$, is obtained by $p_{\mathrm{qfull}}(r_y) = \boldsymbol{\pi}_M(r_y)\boldsymbol{A}_u\mathbf{1}$, where $\boldsymbol{\pi}_M(r_y)$ is the probability vector corresponding to the layer M in $\boldsymbol{\pi}(r_y)$. In (25), $p_{\mathrm{fw}}(\boldsymbol{y}, \boldsymbol{x})$ is proportional to the probability that node $\boldsymbol{x}$ is *available* when $\boldsymbol{y}$ transmits a beacon, and is normalized on the total probability of availability for all possible nodes. The probability of availability is given by

$$p_{\mathrm{avail}}(\boldsymbol{y}, \boldsymbol{x}) = p_{\mathrm{ex}}(r)p_{\mathrm{wake}}(r)p_{\mathrm{SNR}}(\boldsymbol{y}, \boldsymbol{x}), \tag{27}$$

where $p_{\mathrm{wake}}(r) = \sum_{j=1}^{L_w} \boldsymbol{\pi}_{W_j}(r)$ is the probability that node $\boldsymbol{x}$ is awake, and $\boldsymbol{\pi}_{W_j}(r)$ is the equilibrium probability that node $\boldsymbol{x}$ is in the jth active state in $\{X_n\}$. Then, $p_{\mathrm{fw}}(\boldsymbol{y}, \boldsymbol{x})$ in (25) is calculated as

$$p_{\mathrm{fw}}(\boldsymbol{y}, \boldsymbol{x}) = \frac{p_{\mathrm{avail}}(\boldsymbol{y}, \boldsymbol{x})}{\sum_{z \in \mathbb{F}(\boldsymbol{y})} p_{\mathrm{avail}}(\boldsymbol{y}, \boldsymbol{z})}, \tag{28}$$

where node $\boldsymbol{z}$, with the polar coordinates (r_z, β), can be in any small area in $\mathbb{F}(\boldsymbol{y})$.

Thus, according to (25), the traffic rate of node $\boldsymbol{x}$ at each state is determined. Accordingly, $\{I_n\}$ and $\{C_n\}$ are characterized by:

- The (v, v')th element in $\boldsymbol{P}_I$ and $\boldsymbol{P}_C$ is the transition probability from state v to v' shown in Fig. 5.
- The element in $\boldsymbol{\alpha}_I$ and $\boldsymbol{\alpha}_C$ is 1 for states denoted by a "begin" arrow. Other elements are 0's.
- The element in $\boldsymbol{t}_I^{\mathrm{s}}$, $\boldsymbol{t}_C^{\mathrm{s}}$, and $\boldsymbol{t}_C^{\mathrm{f}}$ is set according to the probability attached to the arrows denoted by "success" and "fail", respectively.
- The elements in $\boldsymbol{\lambda}_I$ that correspond to the sleeping states, and the elements in $\boldsymbol{\lambda}_C$ are set to λ_{lc}. Other elements in $\boldsymbol{\lambda}_I$ are set to $\lambda_{\mathrm{lc}} + \lambda_{\mathrm{re}}(r)$.

Then, the equilibrium state probability vector, $\boldsymbol{\pi}(r)$, for the DTMC $\{X_n\}$ is obtained for each node $\boldsymbol{x}$. Consequently, the single-hop delay distribution and end-to-end delay distribution for each ring are obtained according to (13) and (20), respectively.

Next, we focus on empirical experiments to evaluate the analysis model for anycast protocols.

3.6 Experiments

The discussed analytical model can be used to effectively study the distribution of end-to-end delay distribution. It is essential to evaluate the analysis model using empirical studies and define the limitations of the model. In the following, we provide some insight into the model through experiments. The end-to-end delay distribution model is evaluated using MATLAB to determine the single-hop and multi-hop delay distributions [68, 71]. Moreover, empirical experiments and TOSSIM-based simulations are discussed. The default radio and timing parameters of the experiments are listed in Table 1, and the parameters for the channel model are listed in Table 2.

We first show that the analytical results of the end-to-end delay distribution are validated by the simulation and the testbed experiments. The anycast protocol described in Sect. 3.5 is implemented in TinyOS 2.0. The evaluation testbed consists of 25 Crossbow TelosB motes. The nodes are randomly placed in a circular area of radius $R = 4.5\,\text{m}$, where the density is roughly $\rho = 0.39$. Each node generates the same amount of local traffic to be sent to the sink according to a Bernoulli process with average rate $\lambda_{\text{lc}} = 0.001$ in each time unit $T_{\text{u}} = 0.01\,\text{s}$, which equals to 0.1 packet per second. The default duty cycle is $\boldsymbol{x} = 0.5$. The simulation is performed on the same topology. Both the simulations and the testbed experiments have been run for 2.5 h and the end-to-end delay distribution for a node at distance $r = 4.3\,\text{m}$ is recorded, respectively. Other parameters are shown in Table 3.

When compared with the simulations and experiments, as shown in Fig. 7a, it is observed that the analytical results agree well with both results with an the error of less than 10 %. With this accuracy, we can focus on simulations to analyze networks larger than 25 nodes.

Table 1 List of radio and timing parameters for TinyOS CSMA/CA protocol

Group	Notation	Description	Default value
Radio	l_{p}	Data packet size	40 bytes
	R_{b}	Channel bit rate	250 kbps
	P_{t}	Transmit power	−15 dBm
Timing	T_{u}	Time unit	320 μs
	$T_{\text{ibo}}^{\max}$	Maximum initial backoff	9.77 ms
	$T_{\text{cbo}}^{\max}$	Maximum congestion backoff	2.44 ms

Table 2 List of channel-related constants and parameters

Group	Notation	Description	Default value
Channel	P_{n}	Noise floor	−105 dBm
	$PL(D_0)$	Pass loss at reference distance	52.1 dB
	D_0	Reference distance	1 m
	η	Pass loss exponent	3.3
	σ^{s}	Standard deviation of log-normal fading/shadowing	5.5

Table 3 List of parameters for the anycast protocol

Group	Notation	Description	Default Value
Radio	l_p	Data packet size	40 bytes
	R_b	Channel bit rate	250 kbps
	P_t	Transmit power	−15 dBm
	l_m	Beacon and CTS message size	22 bytes
Timing	T_p	Duty cycle period	1 s
	T_a	Active period	0.5 s
	T_b	Beacon transmission timeout	1 s
	T_{to}	Beacon transmission interval	12 ms
	T_u	Time unit	0.01 s
	T_{ibo}^{max}	Maximum initial backoff	9.77 ms
	T_{cbo}^{max}	Maximum congestion backoff	2.44 ms
Protocol	r_{th}	Threshold radius	2.7 m
	ψ_{th}	Threshold SNR	10 dBm

Now, we look at a larger network setting, where the network radius is set to 50 m and the transmission power is increased to −10 dBm. Accordingly, the threshold distance is changed to $r_{th} = 10$ m. Moreover, the network density is $\rho = 0.1$. Durations T_p, T_a, and T_b are 10, 5 and 10 s, respectively, and the traffic rate is 0.01 pkt/sec. Other parameters are left unchanged. 20 different topologies are randomly generated according to a Poisson distribution with the same density. Each topology is simulated for an hour. The end-to-end delay distribution from all nodes with a distance of 50 m to the sink is shown in Fig. 7b, along with the analytical results. It can be observed that the analytical result is also within an error of 10 % of the simulation result.

For any network setup in the experiments, the calculation for the end-to-end delay distribution during any given duration takes less than 2 min. On the other hand, the TOSSIM-based simulations determine the delay distribution in the same order of actual time. For example, for a simulated duration of 2 h, the simulation takes roughly 30 min. Thus, the analytical approach provides insights significantly faster.

4 Event Detection Delay Distribution

So far, we have focused on the distribution of delay of *one packet* as it traverses in a multi-hope network. In typical event monitoring applications, however, numerous sensor nodes are deployed in the space, and operate collaboratively to monitor, report, and react to various physical events. In most WSN data monitoring applications, *events* of interest are detected by sensor nodes, and the user is interested in these events that can be detected by *multiple packets* being reported to a sink via multi-hop communication. The event detection delay consists of *discovery delay* for individual nodes to sense and detect the event, and the *delivery delay* for the network to relay

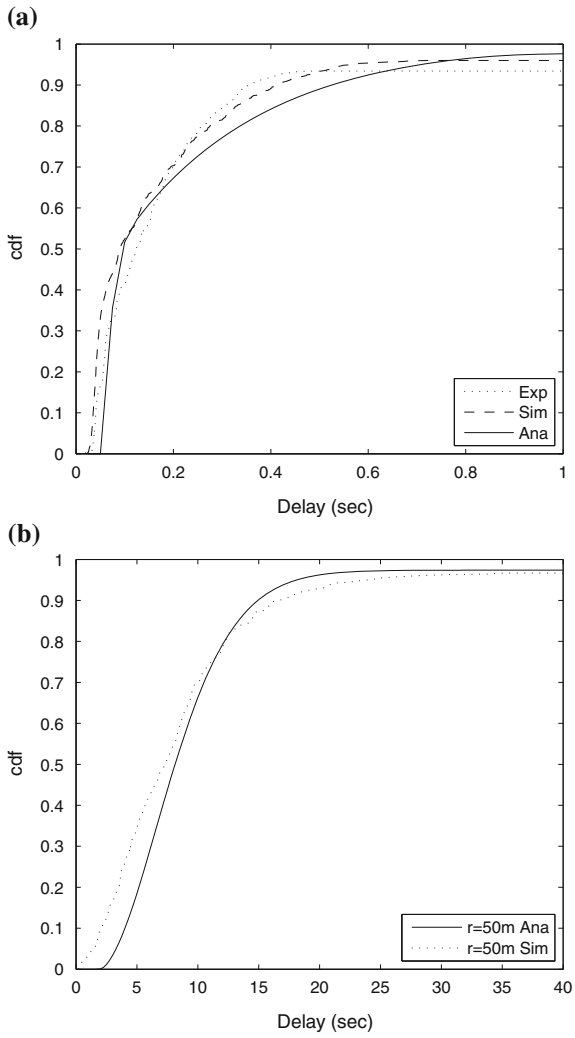


Fig. 7 **a** The analysis, simulation, and experiment results of end-to-end delay distribution with the Anycast protocol for a node with distance $r = 4.3\,\text{m}$ to the sink. **b** The analysis and simulation results for a distance $r = 50\,\text{m}$ to the sink

reports to the sink. When a given number, n, of packets are received by the sink, the event is considered to be detected. Analyzing the event detection delay is a crucial task for real-time WSN applications, which require predictable event detection delay guarantees to be provided by the network.

In this section, we discuss the distribution of event detection delay in WSNs [70]. A spatio-temporal fluid model is presented to derive the distribution of event detection delay. Accordingly, the mean event detection delay and soft-delay bounds for event detection can be modeled. We also show that motivated by the fact that queue build up in low-rate traffic is negligible, a lower-complexity model can be developed. This model extends the delay analysis for single packets, and derives the event detection delay by first obtaining the end-to-end delay for each packet.

4.1 Background

Event detection delay is associated with *flows* of packets in the network. Historically, characterizing timing performance for traffic flows has been investigated in different contexts. Several models have been developed to analyze probabilistic *bounds* on the delay of traffic flows. As an example, the concept of Network Calculus [15] is extended to derive probabilistic bounds for delay through worst case analysis [9, 22]. However, due to the randomness in and the low power nature of the communication links in WSNs, these worst case bounds cannot capture the stochastic characteristics of end-to-end delay. The communication capacity bounds for wireless networks are investigated [20, 23, 29, 43, 73] with saturated traffic flows. In WSNs, the wireless channel utilization is often well below the transmission capacity as nodes are constantly forced into a sleeping state to preserve energy.

In IP network analysis, fluid-based models have been widely exploited [36, 42], and these models have recently been utilized in the analysis of WSNs [19]. Motivated by the fact that the individual packet behavior is less significant when a flow is concerned, the traffic is considered as a continuous flow instead of individual packets in these models. Accordingly, the complexity of the model can be greatly reduced. Furthermore, spatial fluid-based models have also been utilized recently to model stationary properties [13, 64], such as traffic rate and energy consumption for large-scale WSNs. These models greatly reduce the complexity of the (otherwise intractable) problem in either temporal or spatial domains. The discussion that follows builds on top these building blocks for the analysis of event detection delay.

So far, event detection delay in WSNs is analyzed in two main contexts: (1) the event discovery delay, i.e., the delay until the event is detected by an individual node [10, 17], or (2) the delivery delay in a broadcast network [27]. These models, unfortunately, cannot be easily employed for large-scale and multi-hop WSNs, where the models are intractable.

4.2 Problem Definition and System Model

Let us again assume a network deployment, where nodes are randomly located according to a Poisson point process and the node density is ρ. A sink node is deployed at location $\boldsymbol{s} = (x_s, y_s)$, as shown in Fig. 8. At time $t = t_0$, a physical

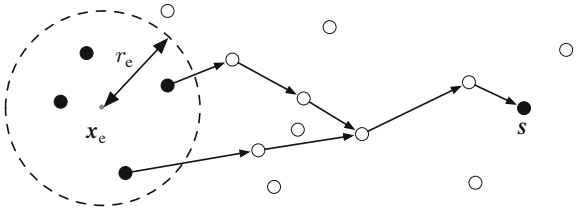


Fig. 8 The network including the sink and the event generation area

event occurs at location $\boldsymbol{x}_e = (x_e, y_e)$, which is called the event center, and lasts for duration T_e. As shown in Fig. 8, all sensor nodes within the detection range, r_e, can discover the event. Each sensor node periodically measures the physical world every t_e seconds using its attached sensors. During the event duration $[t_0, t_0 + T_e)$, whenever the value of the measurement satisfies a predefined rule, e.g., temperature higher than a given threshold, a report packet of size L is generated and is forwarded to the sink using an anycast protocol. Due to inherent noise in the sensor readings, n ($n \geq 1$) readings from multiple sensor nodes are required at the sink to successfully detect the event occurrence. Accordingly, we define the following:

Definition 1 *An event is* **n-*detected*** *if n report packets for that event are received by the sink.*

The delay characteristics of event detection can then be modeled based on the following definitions:

Definition 2 *The* **n-*delay*** *of an event is the delay between when the physical event occurs and when the event is n-detected.*

Definition 3 *The* (**p**, **n**)-***delay bound*** *of an event is delay within which the event is n-detected with probability p.*

To evaluate the delay characteristics of event detection in WSNs, given network and protocol parameters, n and p, we are interested in the following problems:

- What is the n-delay distribution of an event?
- What is the average n-delay of an event?
- What is the (p, n)-delay bound of an event?

In Sects. 4.3 and 4.4, we discuss spatio-temporal fluid models to address these questions.

4.3 Transient Analysis

Instead of capturing each individual node, the network can be represented by a continuous fluid entity distributed in the entire network area. Similarly, instead of individual packets, the traffic can be considered as a continuous packet fluid. By utilizing a spatio-temporal fluid model, the complexity of the problem in both spatial and temporal domains can then be reduced, and the problem becomes tractable. Testbed and simulation evaluations (Sect. 4.5) reveal that this approximation still preserves the accurately of the model.

Consider a location in the network area denoted by $\boldsymbol{x} = (x, y)$. We consider the nodes as a fluid entity over the entire space. Then, in an infinitesimal area around location $\boldsymbol{x}$ with size $\mathrm{d}\boldsymbol{x}$,[2] the amount of nodes is $\rho \mathrm{d}\boldsymbol{x}$, where ρ is the node density.

[2] With a slight abuse of denotation, this infinitesimal area is henceforth denoted by $\mathrm{d}\boldsymbol{x}$.

We also denote the *feasible region* of $\boldsymbol{x}$ as $\mathbb{F}_{\boldsymbol{x}}$, and the *infeasible region* of $\boldsymbol{x}$ as $\mathbb{B}_{\boldsymbol{x}}$. To describe the fluid traffic in the spatial fluid network, we begin by introducing the following traffic concepts:

Definition 4 *The generated, incoming, and outgoing traffic rate density for an infinitesimal area* $\mathrm{d}\boldsymbol{x}$ *is respectively defined as the average number of packets generated, received, and transmitted by the nodes within the area, if any, in an infinitesimal duration* $\mathrm{d}t$*, divided by the duration* $\mathrm{d}t$*, and the size of the area* $\mathrm{d}\boldsymbol{x}$.

In other words, the traffic rate densities define the speed at which packets are generated, received, and transmitted in unit space, respectively. In the transient analysis, their values change over time, and thus, are functions of t. The generated, incoming, and outgoing traffic rate density are denoted by $g_{\boldsymbol{x}}(t)$, $\lambda_{\boldsymbol{x}}(t)$, and $\omega_{\boldsymbol{x}}(t)$, respectively. Note that by assuming a fluid model, the *amount of nodes* in an infinitesimal area $\mathrm{d}\boldsymbol{x}$, and the *amount of packets* sent in an infinitesimal duration $\mathrm{d}t$, are not necessarily an integer number.

Definition 5 *The buffered traffic density for an infinitesimal area* $\mathrm{d}\boldsymbol{x}$ *is defined as the average number of packets buffered in the queue by the nodes within the area divided by the size of the area* $\mathrm{d}\boldsymbol{x}$.

The buffered traffic density is also a function of t, and is denoted by $q_{\boldsymbol{x}}(t)$.

In the following, we derive the set of equations that describe the fluid traffic characteristics of the network after $t = t_0$. Without loss of generality, let $t_0 = 0$. For each node, the generated traffic rate density is given by

$$g_{\boldsymbol{x}}(t) = \begin{cases} \dfrac{\rho}{t_{\mathrm{e}}}, & |\boldsymbol{x} - \boldsymbol{x}_{\mathrm{e}}| < r_{\mathrm{e}}, \text{ and } 0 \le t < T_{\mathrm{e}}, \\ 0, & \text{otherwise}, \end{cases} \tag{29}$$

where ρ is the density, t_{e} is the reporting interval, and $|\boldsymbol{x} - \boldsymbol{x}_{\mathrm{e}}|$ denotes the Euclidean distance between $\boldsymbol{x}$ and $\boldsymbol{x}_{\mathrm{e}}$. During an infinitesimal duration $\mathrm{d}t$, the amount of arriving traffic, along with the traffic already stored in the queue is

$$a_{\boldsymbol{x}}(t) = q_{\boldsymbol{x}}(t) + (\lambda_{\boldsymbol{x}}(t) + g_{\boldsymbol{x}}(t)) \cdot \mathrm{d}t \tag{30}$$

which is the available traffic that needs to be transmitted.

For each infinitesimal area $\mathrm{d}\boldsymbol{y}$ in the feasible forwarding region $\mathbb{F}_{\boldsymbol{x}}$ of $\boldsymbol{x}$, the amount of nodes with good channel quality is

$$c_{\boldsymbol{x},\boldsymbol{y}} = \rho \cdot p_{\boldsymbol{x},\boldsymbol{y}}(\psi_{\mathrm{th}}), \tag{31}$$

where $p_{\boldsymbol{x},\boldsymbol{y}}(\psi_{\mathrm{th}})$ is the probability that the CTS message sent from a node at $\boldsymbol{y}$ has a higher SNR than a given threshold ψ_{th} when received by the node at $\boldsymbol{x}$ ((10) in [76]). Thus, the total amount of nodes in $\mathbb{F}_{\boldsymbol{x}}$ with good channel quality is

$$c_{\mathbb{F}_{\boldsymbol{x}}} = \int_{\mathbb{F}_{\boldsymbol{x}}} c_{\boldsymbol{x},\boldsymbol{y}} \mathrm{d}\boldsymbol{y}. \tag{32}$$

Note that between any pair of nodes, at most one packet can be transmitted in a cycle T_p. Thus the maximum amount of traffic transmitted during a cycle T_p from $\mathrm{d}\boldsymbol{x}$ to anywhere in $\mathbb{F}_{\boldsymbol{x}}$ is

$$\Omega_{\boldsymbol{x}}^{\max} = \rho \mathrm{d}\boldsymbol{x} \cdot c_{\mathbb{F}_{\boldsymbol{x}}}. \tag{33}$$

Since the traffic is considered as a fluid and a packet takes one cycle to be transmitted between a pair of nodes, in $\mathrm{d}t$, the maximum amount of traffic sent from $\mathrm{d}\boldsymbol{x}$ is $\Omega_{\boldsymbol{x}}^{\max} \cdot \mathrm{d}t / T_\mathrm{p}$. In the case where each node in $\mathrm{d}\boldsymbol{x}$ has less than 1 available packet in its queue, i.e., $a_{\boldsymbol{x}}(t) < 1 \cdot \rho$, it still takes an entire cycle to transmit them. In this case the actual transmitted traffic during $\mathrm{d}t$ is $\frac{a_{\boldsymbol{x}}(t)}{1 \cdot \rho} \cdot \Omega_{\boldsymbol{x}}^{\max} \cdot \frac{\mathrm{d}t}{T_\mathrm{p}}$. Accordingly, the transmitted traffic rate density at $\boldsymbol{x}$ is

$$\omega_{\boldsymbol{x}}(t) = \min\left[1, \frac{a_{\boldsymbol{x}}(t)}{1 \cdot \rho}\right] \Omega_{\boldsymbol{x}}^{\max} \frac{\mathrm{d}t}{T_\mathrm{p}} \cdot \frac{1}{\mathrm{d}\boldsymbol{x}\mathrm{d}t} = \min\left[a_{\boldsymbol{x}}(t), \rho\right] \frac{c_{\mathbb{F}_{\boldsymbol{x}}}}{T_\mathrm{p}}, \tag{34}$$

where $a_{\boldsymbol{x}}(t)$ is the available traffic density given by (30).

The outgoing traffic in each infinitesimal area is equally distributed to every node with good channel quality in its feasible region. Thus, the incoming traffic rate density, $\lambda_{\boldsymbol{x}}(t)$, that is received from each infinitesimal area in the backward region, $\mathbb{B}_{\boldsymbol{x}}$, is given by

$$\lambda_{\boldsymbol{x}}(t) = \int_{\mathbb{B}_{\boldsymbol{x}}} \omega_{\boldsymbol{y}}(t) \cdot \frac{c_{\boldsymbol{y},\boldsymbol{x}}}{c_{\mathbb{F}_{\boldsymbol{y}}}} \mathrm{d}\boldsymbol{y}. \tag{35}$$

Within duration $\mathrm{d}t$, the change in buffered traffic density is

$$\mathrm{d}q_{\boldsymbol{x}}(t) = \Big(g_{\boldsymbol{x}}(t) + \lambda_{\boldsymbol{x}}(t) - \omega_{\boldsymbol{x}}(t)\Big)\mathrm{d}t, \tag{36}$$

and the buffered traffic density at time $t + \mathrm{d}t$ changes to

$$q_{\boldsymbol{x}}(t + \mathrm{d}t) = q_{\boldsymbol{x}}(t) + \mathrm{d}q_{\boldsymbol{x}}(t) \tag{37}$$

Thus, (34), (35), (36) and (37) describe the traffic dynamics of the network after $t = t_0$. Given the initial value of $q_{\boldsymbol{x}}(t_0)$, the traffic rates in the network can be evaluated for any time instance $t > t_0$. Accordingly, the total incoming traffic rate at the sink, which models the total number of packets received by the sink, can be obtained.

Within the transmission range of the sink, the outgoing traffic rate density in (34) becomes

$$\omega_{\boldsymbol{x}}(t) = a_{\boldsymbol{x}}(t), \tag{38}$$

since the sink is always awake and the traffic can all be transmitted to the sink directly. Moreover, for these nodes, in (35), the backward region $\mathbb{B}_{\boldsymbol{x}}$ excludes the areas within the transmission range of the sink. Then, at the sink, the incoming traffic rate is calculated as

$$\Lambda(t) = \int_{\boldsymbol{x}:|\boldsymbol{x}-s|\leq r_{\text{th}}} \omega_{\boldsymbol{x}}(t)\mathrm{d}\boldsymbol{x}, \tag{39}$$

where r_{th} is the distance threshold around the sink within which all nodes directly send packets to the sink.

To calculate the incoming traffic rate at the sink, the entire network area can be discretized into small areas, where the time is also divided into small time steps. Initially, the buffered traffic density for every infinitesimal area in the network at time $t = 0$ is $q^0_{\boldsymbol{x}}$. $\lambda_{\boldsymbol{x}}(t)$ and $\omega_{\boldsymbol{x}}(t)$ are set as 0. Then, $\omega_{\boldsymbol{x}}(t)$ and $\lambda_{\boldsymbol{x}}(t)$ are calculated using (34) and (35), respectively. Then, $q_{\boldsymbol{x}}(t)$ is updated for the next time step according to (36). This process is repeated for each time step, and $\Lambda(t)$ as a function of t is obtained.

To obtain the n-delay distribution from $\Lambda(t)$, we consider the traffic arrival process to be a Poisson process with variable rate according to $\Lambda(t)$. This model is also observed in empirical evaluations. Consequently, the n-delay distribution, $f_n(t)$, of the non-homogeneous Poisson process is given by [24, Ch. 2.4]:

$$f_n(t) = \frac{[\hat{\Lambda}(t)]^{(n-1)}\Lambda(t)\mathrm{e}^{-\hat{\Lambda}(t)}}{(n-1)!}, \tag{40}$$

where $\hat{\Lambda}(t)$ is the integral of $\Lambda(t)$ over duration $(0, t]$.

Accordingly, we have the following theorem:

Theorem 2. *For a WSN system described in Sect. 4.2, the average n-delay and the* (p, n)*-delay bound of an event are*

$$\mu(n) = \int_0^\infty t f_n(t)\mathrm{d}t, \tag{41}$$

$$j(p, n) = f_n^{-1}(p), \tag{42}$$

respectively, where $f_n(t)$ *is given by (40).*

Proof. Since $f_n(t)$ in (40) is the *pdf* of the n-delay, (41) and (42) are directly obtained according to the definition of the *pdf*.

4.4 Simplified Delay Model

The spatio-temporal fluid model decreases the complexity of the solution significantly. To further reduce the complexity, the low traffic in WSNs can be put to use.

In this simplified model, the network area is divided into small rings. Thus, the spatial calculation complexity is also reduced from 2D to 1D.

For a low traffic rate WSN, the queueing effect can be neglected. Moreover, the channel aware anycast operation allows one to assume that the channel error is negligible within a transmission range of R. For a node located at $\boldsymbol{x}$, after it receives a packet (locally generated or forwarded), in the duration t, the probability that there is no node in its feasible forwarding region $\mathbb{F}_{\boldsymbol{x}}$ waking up is

$$\begin{aligned} p_{\boldsymbol{x}}^{\mathrm{nf}}(t) &\approx \prod_{\boldsymbol{y}=(l,\theta)\in\mathbb{F}_{\boldsymbol{x}}} \left[1-\rho \mathrm{d}\boldsymbol{y}\, p^{\mathrm{wake}}(t)\right] = \left[1-\rho \mathrm{d}\boldsymbol{y}\, p^{\mathrm{wake}}(t)\right]^{\frac{A_{\mathbb{F}_{\boldsymbol{x}}}}{\mathrm{d}\boldsymbol{y}}} \\ &= \exp\left(-A_{\mathbb{F}_{\boldsymbol{x}}}\rho p^{\mathrm{wake}}(t)\right), \end{aligned} \tag{43}$$

where the product is over $\mathbb{F}_{\boldsymbol{x}}$, divided according to the polar coordinates originated at $\boldsymbol{x}$, ρ is the network density, $A_{\mathbb{F}_{\boldsymbol{x}}}$ is the size of $\mathbb{F}_{\boldsymbol{x}}$, and $p^{\mathrm{wake}}(t)$ is the probability that a node in each region wakes up during the period t. Since the wake period of each node is unsynchronized with each other, $p^{\mathrm{wake}}(t)$ is irrelevant to the location. Moreover, since each node wakes up at uniformly distributed times, we have

$$p^{\mathrm{wake}}(t) = \begin{cases} \dfrac{t}{T_{\mathrm{p}}}, & 0 \le t \le T_{\mathrm{p}} \\ 1, & t > T_{\mathrm{p}} \end{cases}. \tag{44}$$

Therefore, the probability that at least one node in $\mathbb{F}_{\boldsymbol{x}}$ wakes up during t is

$$p_{\boldsymbol{x}}^{\mathrm{fwake}}(t) = 1 - p_{\boldsymbol{x}}^{\mathrm{nf}}(t) = \begin{cases} 1-e^{-A_{\mathbb{F}_{\boldsymbol{x}}}\rho t/T_{\mathrm{p}}}, & 0 \le t \le T_{\mathrm{p}} \\ 1-e^{-A_{\mathbb{F}_{\boldsymbol{x}}}\rho}, & t > T_{\mathrm{p}} \end{cases}. \tag{45}$$

This is exactly the *cdf* of the single hop delay. Therefore, the *pdf* of single hop delay for a node at $\boldsymbol{x}$ is

$$f_{\mathrm{sh}(\boldsymbol{x})}(t) = \mathrm{d}p_{\boldsymbol{x}}^{\mathrm{fwake}}(t)/\mathrm{d}t = \begin{cases} \dfrac{A_{\mathbb{F}_{\boldsymbol{x}}}\rho}{T_{\mathrm{p}}} e^{-A_{\mathbb{F}_{\boldsymbol{x}}}\rho t/T_{\mathrm{p}}}, & 0 \le t \le T_{\mathrm{p}} \\ 0, & t > T_{\mathrm{p}} \end{cases}. \tag{46}$$

The end-to-end delay distribution from location $\boldsymbol{x}$ to the sink can be found as the convolution of single-hop delay distributions in the path as explained in Sect. 3. Thus, the *pdf* of end-to-end delay from $\boldsymbol{x}$ to the sink is

$$f_{\mathrm{e2e}(\boldsymbol{x})}(t) = \int_{\boldsymbol{y}\in\mathbb{F}_{\boldsymbol{x}}} f_{\mathrm{e2e}(\boldsymbol{x}')} * f_{\mathrm{sh}(\boldsymbol{x})}(t)\rho \mathrm{d}\boldsymbol{y}, \tag{47}$$

where $\boldsymbol{y}$ is the location (l,θ). Since the queueing effect is neglected, the nodes with the same distance to the sink have the same end-to-end delay to the sink. Therefore,

the end-to-end delay distribution is calculated only once for all nodes with the same distance to the sink. This fact results in a significant reduction on the calculation time.

Suppose the packet generation function for a node at $\boldsymbol{x}$ is $g_{\boldsymbol{x}}(t)$, then the packet reception rate from $\boldsymbol{x}$ by the sink is

$$\lambda_{\boldsymbol{x}}(t) = g_{\boldsymbol{x}} * f_{\mathrm{e2e}(\boldsymbol{x})}(t). \tag{48}$$

Then, the packet reception rate at the sink is the sum of traffic generated from each location in the event detection region. Thus,

$$\Lambda(t) = \int_{\boldsymbol{x}\in\mathbb{E}} g_{\boldsymbol{x}} * f_{\mathrm{e2e}(\boldsymbol{x})}(t)\mathrm{d}\boldsymbol{x}, \tag{49}$$

where $\mathbb{E}$ is the region within the detection range, r_{e}, of the event location, $\boldsymbol{x}_{\mathrm{e}}$, i.e., $\mathbb{E} = \{\boldsymbol{x} : |\boldsymbol{x} - \boldsymbol{x}_{\mathrm{e}}| \leq r_{\mathrm{e}}\}$.

Finally, the distribution of event detection delay is obtained by using (49) in (40), and the average n-delay and the (p, n)-delay bound of an event are obtained by Theorem 2.

4.5 Experiments

Now, we evaluate the accuracy of the both models using testbed experiments and simulations. The average n-delay and the (p, n)-delay bound of an event in the experiments and simulations are used to compare against the framework. The spatio-temporal fluid model in the framework is implemented using C++ and the simplified model is implemented using MATLAB.

Again, using testbed experiments, we can evaluate the effectiveness of the models in small-scale deployments. Using a similar setup as in Sect. 3, experiments are performed. In addition, we also discuss simulation results using TOSSIM [39] with the same parameters.

The results are shown in Fig. 9a along with the results given by the two analytical models in (39) and (49). As can be seen in Fig. 9a, both testbed experiments and simulations validate the models. The (p, n)-delay bound for $p = 0.75$ is also calculated for different n's. The results are shown in Fig. 9b. For the majority of the cases, testbed and simulation results are within 5 % of the model. Moreover, the results also confirm the accuracy of TOSSIM simulations, which are used in further evaluations of the two models in larger-scale networks. The results show that the model is accurate for low traffic high density WSN cases.

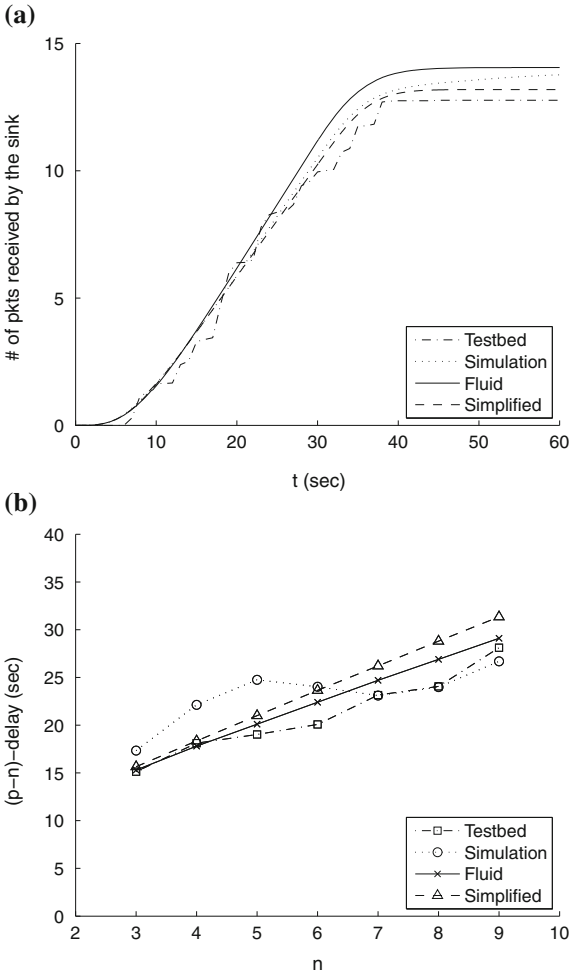


Fig. 9 The event delay results of testbed experiment, simulation, and models (**a**) The reception rate at the sink (**b**) The (p;n)-delay bound of the event delay for p = 0:75

4.5.1 Comparison Between the Models

The spatio-temporal model and the simplified fluid model provides different advantages and disadvantages for network modeling. Both models yield the results in a significantly less time than simulations. For a typical network of 400 nodes, the simulation takes more than one day to complete, while the complete fluid model takes around 10 min to calculate, and the simplified only takes less than 1 min. It is shown in Fig. 9 that both models yield accurate results. The only cases where the result is less accurate is when the density is low, or when the traffic rate is high, which are not common configurations for WSNs.

Compared to the spatio-temporal model, the simplified model decreases the complexity at the cost of accuracy. The simplified model requires $O(\sqrt{A})$ time, where A

is the area of the network. On the other hand, since the spatio-temporal fluid model calculates the traffic rates for each location in the entire 2D network, it requires $O(A)$ time. On the other hand, the simplified model is less accurate when the nodes with the same distance to the sink do not have the same end-to-end delay. An example is a non-regular network where nodes density varies over the space. The complete fluid model, however, can be extended to provide accurate result in such networks when the density ρ in (29)–(34) by $\rho(\boldsymbol{x})$, a density function of corresponding location $\boldsymbol{x}$.

5 Network Lifetime Distribution

Now we turn our attention to energy consumption and lifetime in WSNs. Due to similar reasons, energy consumption also has a stochastic characteristic and its distribution characteristics should be analyzed. It is important to predict and guarantee the lifetime of a network before it is deployed for practical reasons.

We start by modeling the distribution of energy consumption at the node level and extend this analysis to the network level. Accordingly, lifetime of a node and the whole network is modeled using stochastic measures. The analysis in this section is based on the models presented in Sect. 3, since the analysis of both delay and energy consumption utilize the Discrete-Time Markov model at the node level. Moreover, we show that for large enough time durations, energy consumption *converges to a Normal distribution*. This result greatly reduces the computation cost for the analysis. We again use the anycast protocol as a case study to illustrate the use of the model and discuss validations with experiments and simulations.

5.1 Background

Historically, energy consumption and lifetime has mainly been analyzed in the average sense. Average energy efficiency is evaluated for specific protocols [8, 57, 74], and average energy consumption models are developed [16, 48]. The effects of routing strategies on energy consumption have also been investigated recently using a realistic radio model is adopted to analyze the tradeoff between single-hop long-distance transmissions and multi-hop short-distance transmissions [30]. The same problem is also investigated using an energy model focused on circuit level hardware [67]. Stochastic characteristics of energy consumption, however, cannot be captured by these models.

Lifetime of WSNs has also been investigated predominantly using average measures [18, 32, 37]. The probabilistic lifetime analysis of a cluster-based network for a TDMA MAC protocol [53] is a first step towards higher order statistics and here we provide a stochastic energy consumption and network lifetime model for multi-hop WSNs.

5.2 Problem Definition and System Model

In WSNs, energy is consumed by each node for various activities including sensing, data processing, and communication. We assume that each node is equipped with K sensors, and each sensor $k \in \{1, \ldots, K\}$ is used to sense the physical environment every $T_{s,k}$ seconds (subscript "s" refers to "sensing") with an energy consumption of $\varepsilon_{s,k}$. Based on the application requirements, a packet is generated locally if the sensed information satisfies event definitions. For each received and locally generated packet, the node processes the data with an energy consumption of ε_p. Moreover, the energy consumption for the communication, ε_c, is a variable dependent on the network parameters and the protocols running on each node.

For a given random network topology, we are interested in the following problems:

1. Given a period of time T, what is the energy consumption distribution, $F_{E(\boldsymbol{x},T)}(e)$, of a node at $\boldsymbol{x}$?
2. Given the energy consumption distribution, what is the lifetime distribution, $F_{L(\boldsymbol{x},C(\boldsymbol{x}))}(t)$, of a node at $\boldsymbol{x}$?
3. Given the energy consumption distribution for each node $\boldsymbol{x}$ in the network, what is the distribution of the network lifetime, $F_{NL}(t)$?

Next let us provide an overview of the solutions for the above problems. The details of the models are then elaborated in Sects. 5.3 and 5.4.

5.2.1 Single Node Energy Consumption Distribution

The randomness in energy consumption and the associated lifetime is due to the randomness because of the wireless channel errors and queueing operation. Accordingly, for a node at $\boldsymbol{x}$, the r.v. for energy consumption during a given time period, T, is expressed as:

$$E(\boldsymbol{x}, T) = E_s(\boldsymbol{x}, T) + E_{cp}(\boldsymbol{x}, T), \tag{50}$$

where $E_s(\boldsymbol{x}, T)$ is the r.v. of energy consumption for sensing, and $E_{cp}(\boldsymbol{x}, T)$ is the r.v. of energy consumption for communication and processing. These two terms capture the randomness due to protocol operation.Accordingly, the *pdf* of the total energy consumption of a node at $\boldsymbol{x}$ is

$$f_{E(\boldsymbol{x},T)}(e) = f_{E_s(\boldsymbol{x},T)} * f_{E_{cp}(\boldsymbol{x},T)}, \tag{51}$$

where the *pdf* of the corresponding r.vs. in (50) are convolved.

(1) Energy Consumption for Sensing: During any given time duration T starting at t_0, i.e., the period $[t_0, t_0 + T)$, for some sensor k with periodic sensing interval $T_{s,k}$ and energy consumption per sensing $\varepsilon_{s,k}$, denote the first sensing activity for sensor k after t_0 occurs at t_{k1}. The number of sensing activities during T is then $n_k(T) = \lceil (t_0 + T - t_{k1})/T_{s,k} \rceil$. Since t_0 is independent of sensing activities, $t_{k1} - t_0$

is a r.v. uniformly distributed in range $[0, T_{s,k})$. Therefore, the *pmf* of $n_k(T)$ is given by

$$f_{n_k(T)}(n) = \begin{cases} N_{s,k} - n + 1, & n = \lfloor N_{s,k} \rfloor + 1 \\ n + 1 - N_{s,k}, & n = \lceil N_{s,k} \rceil \\ 0, & \text{otherwise} \end{cases}, \tag{52}$$

where $N_{s,k} = \dfrac{T}{T_{s,k}}$. The *pdf* of energy consumption for all K sensors during T is obtained as

$$\begin{aligned} f_{E_s(T)}(e) &= \sum_{k=1}^{K} \sum_{n} n \cdot \delta(e - f_{n_k(T)}(n)) \\ &= \sum_{k=1}^{K} \Big[\left(N_{s,k} - \lfloor N_{s,k} \rfloor\right) \cdot \delta\left(e - \left(\lfloor N_{s,k} \rfloor + 1\right) \varepsilon_{s,k}\right) \\ &\quad + \left(\lfloor N_{s,k} \rfloor + 1 - N_{s,k}\right) \cdot \delta\left(e - \lfloor N_{s,k} \rfloor \varepsilon_{s,k}\right) \Big]. \end{aligned} \tag{53}$$

(2) Energy Consumption for Communication and Processing: Next, we briefly introduce the model for the analysis of the energy consumption for communication and processing, $E_{cp}(T)$, and leave the details of the model to Sect. 5.3.

The energy consumed by communication and data processing at each node in the network is modeled by a discrete-time queueing system with time unit T_u, which is characterized by its traffic inter-arrival distribution and service process. More specifically, in each time unit, the traffic inter-arrival is modeled according to a Bernoulli process, and a variant of the Discrete Time Markov Process (DTMP) discussed in Sect. 3.3 is used to model the service behavior.

Similar to the DTMP model in Sect. 3.3, the DTMP for the energy consumption analysis is represented by a Discrete-Time Markov Chain (DTMC) $\{X_n\}$. Contrary to the delay analysis, each state v in $\{X_n\}$ is also associated with an amount of energy, ε_v, consumed for the corresponding activity during T_u. The communication and data processing behaviors of each node are represented by transitions among states in $\{X_n\}$. We defer the detailed explanation of this DTMC to Sect. 5.3. Based on this DTMC, the *pdf* of the single-node energy consumption for communication and data processing, $E_{cp}(T)$, is found for any given duration T.

5.2.2 Node Lifetime and Network Lifetime Distributions

For a battery-powered sensor node, its lifetime distribution depends on the energy consumption distribution during any given period T, and the total capacity of its battery C. The network lifetime distribution depends on the lifetime distribution for each node, and how the network lifetime is defined. In the following, we focus on the lifetime defined as follows.

Definition 6 *The network lifetime is defined as the duration before the battery depletion of* ***the first node****.*

5.3 Single Node Energy Consumption Distribution

The energy consumed by communication and data processing for a node is represented by the energy costs associated with transitions among states in Markov chain $\{X_n\}$. In the following, based on the discussion in Sect. 3.3, the construction of states and transitions in $\{X_n\}$ is discussed.

5.3.1 Structure of Markov Chain $\{X_n\}$

According to the MAC protocol employed, the structures of $\{C_n\}_m$ and $\{I_n\}$ are parameterized by the following variables: $\boldsymbol{P}_I$, $\boldsymbol{P}_C$, $\boldsymbol{\alpha}_I$, $\boldsymbol{\alpha}_C$, $\boldsymbol{t}_I^{\mathrm{s}}$, $\boldsymbol{t}_C^{\mathrm{s}}$, $\boldsymbol{t}_C^{\mathrm{f}}$, $\boldsymbol{\lambda}_I$, and $\boldsymbol{\lambda}_C$. The definitions of these variables are given in Sect. 3. Accordingly, the transition probability matrix, $\boldsymbol{Q}_X$, of the entire Markov chain $\{X_n\}$ can be found based on (5). Then, the equilibrium state probability vector, $\boldsymbol{\pi}$, for $\{X_n\}$ is calculated by solving $\boldsymbol{\pi}\,\boldsymbol{Q}_X = \boldsymbol{\pi}$.

5.3.2 Energy Consumption for Communication and Processing

Now, based on the DTMC, we derive the distribution of energy consumption for communication and processing. Suppose at the beginning of a time unit T_{u}, the node is in state v of $\{X_n\}$, and during the time unit, the energy consumption of the node for communication and data processing is ε_v. The value of ε_v is obtained from measurement, or is calculated according to the specifications of the hardware platform. The *cdf* and the *pdf* of $E_{\mathrm{cp}}(T_{\mathrm{u}})$, the energy consumption during the time unit, are $G_v(e) = u(e - \varepsilon_v)$ and $g_v(e) = \delta(e - \varepsilon_v)$, respectively, where $u(\cdot)$ is the unit step function and $\delta(\cdot)$ is the delta function.[3] We denote

$$H^{(1)}_{v,v'}(e) = G_v(e) q_{v,v'} = \Pr\{E_{\mathrm{cp}}(T_{\mathrm{u}}) \le e \cap v \xrightarrow{1} v'\}, \tag{54}$$

$$h^{(1)}_{v,v'}(e) = g_v(e) q_{v,v'} = \mathrm{d}H^{(1)}_{v,v'}(e)/\mathrm{d}e, \tag{55}$$

where $v \xrightarrow{1} v'$ represents the event that $\{X_n\}$ transitions from state v to v' in one time unit, and $q_{v,v'}$ is the (v, v')th element of the transition probability matrix, $\boldsymbol{Q}_X$, in (5).

[3] Although a discrete time Markov process is used for the model, the energy consumption is *continuous*. Thus the *pdf*, as opposed to the *pmf*, is used to characterize the distribution.

For a given period T, the number of time units of T_{u} is $\hat{T} \sim T/T_{\mathrm{u}}$ (since T_{u} is usually chosen to be very small, T is approximated as an integer multiple of T_{u}). After $\hat{T}$ time units ($\hat{T} > 1$), the *cdf* of energy consumption is found to be

$$\begin{aligned} H_{v,v'}^{(\hat{T})}(e) &= \Pr\{E_{\mathrm{cp}}(T) \le e \cap v \xrightarrow{\hat{T}} v'\} = \int_0^e h_{v,v'}^{(\hat{T})}(\epsilon)\mathrm{d}\epsilon \\ &= \int_0^e \sum_{v'' \in \mathscr{S}} (h_{v,v''}^{(1)} * h_{v'',v'}^{(\hat{T}-1)})(\epsilon)\mathrm{d}\epsilon \end{aligned} \tag{56}$$

where $\mathscr{S}$ is the set of all states in $\{X_n\}$. Therefore, if the matrix of $h_{v,v'}^{(\hat{T})}(e)$ is denoted by $\boldsymbol{h}^{(\hat{T})}(e)$, then $\boldsymbol{h}^{(\hat{T})}(e)$ is the $\hat{T}$-fold convolution of $\boldsymbol{h}^{(1)}(e)$.

The energy consumption distribution during T depends on the initial state of the system at the beginning of this period, which is usually randomly chosen. Thus, the initial state probability vector is represented by the equilibrium state probability vector $\boldsymbol{\pi}$. After $\hat{T}$ time units, the *pdf* and the *cdf* of the energy consumption are

$$\begin{aligned} f_{E_{\mathrm{cp}}(T)}(e) &= \boldsymbol{\pi}\boldsymbol{h}^{(\hat{T})}(e)\mathbf{1}, \\ F_{E_{\mathrm{cp}}(T)}(e) &= \int_0^e f_{E_{\mathrm{cp}}(T)}(\epsilon)\mathrm{d}\epsilon, \end{aligned} \tag{57}$$

respectively, where $\mathbf{1}$ is the appropriately dimensioned column vector containing all 1's.

5.3.3 Asymptotic Energy Consumption Distribution

When the given duration T is large, the calculation of energy consumption distribution may require extensive computing power, especially for (57). On the other hand, if a QBD process is modeled by a DTMC, and each state in the DTMC is associated with a cost, then the sum of the total cost during a given period T asymptotically approaches the Normal distribution as $T \to \infty$ [49]. This is the same case for the energy consumption distribution model, which can be approximated by a Normal distribution, whose mean and the variance are given by

$$\lim_{T\to\infty} \mu_{\mathrm{cp}}(T) = \hat{T}\mu_{\mathrm{cp,u}} = \hat{T}\boldsymbol{\pi}\boldsymbol{\varepsilon}, \tag{58}$$

$$\lim_{T\to\infty} \sigma_{\mathrm{cp}}^2(T) = \hat{T}\sigma_{\mathrm{cp,u}}^2 = \hat{T}\left(\sum_{v\in\mathscr{S}} (\varepsilon_v - \boldsymbol{\pi}\boldsymbol{\varepsilon})^2\pi_v + 2\boldsymbol{\beta}\boldsymbol{\varepsilon}\right), \tag{59}$$

respectively, where $\hat{T} = T/T_{\mathrm{u}}$ is the number of time units in T, $\mu_{\mathrm{cp,u}}$, and $\sigma_{\mathrm{cp,u}}^2$ are the mean and variance of E_{cp} during each time unit T_{u}. Moreover, $\boldsymbol{\pi}$ is the equilibrium state probability vector of $\{X_n\}$, $\mathscr{S}$ is the set of states in $\{X_n\}$, and π_v

is the equilibrium state probability for state v. Finally, $\boldsymbol{\varepsilon}$ is the vector of ε_v for each state v in $\{X_n\}$, and $\boldsymbol{\beta}$ is an intermediate vector variable which is obtained by solving the following set of equations [49]:

$$\boldsymbol{\beta}(\boldsymbol{Q}_X - \boldsymbol{I}) = -\boldsymbol{\gamma}\,\boldsymbol{Q}_X, \tag{60}$$

$$\boldsymbol{\beta}\mathbf{1} = 0, \tag{61}$$

where $\boldsymbol{\gamma}$ is a row vector whose v-th element is $(\varepsilon_v - \boldsymbol{\pi}\boldsymbol{\varepsilon})\pi_v$.

Then, the asymptotic distribution for $E_{\mathrm{s}}(T)$, the energy consumption by sensing, is also derived. For each sensor k, when $T \to \infty$, $\left\lfloor \frac{T}{T_{\mathrm{s},k}} \right\rfloor \approx \frac{T}{T_{\mathrm{s},k}} \approx \left\lfloor \frac{T}{T_{\mathrm{s},k}} \right\rfloor + 1$. Thus, (53) becomes

$$f_{E_{\mathrm{s}}(T)}(e) \approx \delta\left(e - \sum_{k=1}^{K} \frac{\varepsilon_k T}{T_{\mathrm{s},k}}\right). \tag{62}$$

In other words, the energy consumption is approximately linear to T with a constant coefficient equal to $\sum_{k=1}^{K} \varepsilon_k / T_{\mathrm{s},k}$.

Therefore, the following results are obtained:

Theorem 3. *When $T \to \infty$, the energy consumption of a node during T asymptotically approaches the Normal distribution, with the mean and variance linear to T and given by*

$$\mu(T) = \hat{T}\left(\mu_{cp,u} + \sum_{k=1}^{K} \frac{\varepsilon_k T_u}{T_{s,k}}\right), \tag{63}$$

$$\sigma^2(T) = \hat{T}\sigma_{cp,u}^2 + cT^2, \tag{64}$$

where $\hat{T} = T/T_u$ is the number of time units in T.

Proof. The proof is trivial by combining (58), (62).

5.4 Lifetime Distribution Analysis

Using the *pdf* of energy consumption $f_{E(T)}(e)$ in (57) for any given period T, the lifetime distribution of a node, and further, the entire network, can be found as follows.

5.4.1 Single-Node Lifetime Distribution

The distribution of the lifetime for a given node, $L(C)$, is a function of its total battery capacity C. Initially, the node has a battery residual of C. After duration T, the *pdf*

of remaining energy in the battery is $f_{C-E(T)}(e)$. The probability that the node has a shorter lifetime than duration T is the probability that the remaining energy after T is lower than 0. Thus, the *cdf* of the node lifetime is

$$F_{L(C)}(T) = \Pr\{L(C) \leq T\} = \Pr\{C - E(T) \leq 0\}. \tag{65}$$

As explained in Sect. 5.3.3, when T is large, $E(T) \sim \mathcal{N}(\mu(T), \sigma^2(T))$, where $\mu(T)$ and $\sigma^2(T)$ are given by Theorem 3. Thus, the *cdf* of single-node lifetime is approximated as

$$F_{L(C)}(t) \approx Q\left(\frac{\mu(t) - C}{\sqrt{\sigma^2(t)}}\right). \tag{66}$$

5.4.2 Network Lifetime Distribution

Since every node needs to be alive during the network lifetime, the network lifetime (NL) distribution is obtained for a WSN with random deployment as:

$$F_{NL}(t) \approx 1 - \prod_{\boldsymbol{x} \in \mathscr{A}} (1 - p_{\text{ex}}(\boldsymbol{x}) \Pr\{L(\boldsymbol{x}, C(\boldsymbol{x})) \leq t\}), \tag{67}$$

where $L(\boldsymbol{x}, C(\boldsymbol{x}))$ is the lifetime for a node located at $\boldsymbol{x}$, if any, with battery capacity $C(\boldsymbol{x})$. Using the approximation in (66) for the single-node lifetime distribution, the network lifetime distribution is approximated by

$$F_{NL}(t) \approx 1 - \prod_{\boldsymbol{x} \in \mathscr{A}} \left(1 - p_{\text{ex}}(\boldsymbol{x}) Q\left(\frac{C(\boldsymbol{x}) - \mu(\boldsymbol{x}, t)}{\sqrt{\sigma^2(\boldsymbol{x}, t)}}\right)\right), \tag{68}$$

where $\mu(\boldsymbol{x}, t), \sigma^2(\boldsymbol{x}, t)$ are given by Theorem 3 for the node located at $\boldsymbol{x}$. Moreover, $\mathscr{A}$ is the network area. To calculate the product, area $\mathscr{A}$ is discretized into small areas of size $\Delta\boldsymbol{x}$, and $p_{\text{ex}}(\boldsymbol{x})$ is the probability that there exist a node in the small area around $\boldsymbol{x}$, and is a function of the network density ρ. It is obtained by $p_{\text{ex}}(\boldsymbol{x}) = \rho\Delta\boldsymbol{x}$.

5.5 Case Study: Anycast Protocol

Using the anycast protocol, now we illustrate how the analysis model can be used for communication protocols. For the energy and lifetime analysis with anycast protocol, we assume that nodes are deployed in a circular plane of radius R, have a homogeneous battery capacity C, and generate a homogeneous amount of local traffic to a sink located at the center of the plane. Because of the symmetry, node-specific variables are the same for each narrow ring with radius r, and are indexed by r. In

the following analysis, when there is no ambiguity, the subscript r in ring-specific variables is omitted.

5.5.1 Energy Consumption in Each State

During the protocol operation,, a node conducts one of the following communication tasks: transmission, listening, receiving, and sleeping. Listening and receiving are considered the same since most popular architectures, such as Mica2 [62] and TelosB [63], consume similar power for these tasks. We also ignore the energy consumed for the data packet transmission. This is a valid simplification because majority of the energy is consumed for idle listening and beacon transmissions. As a result, there are three types of states in $\{X_n\}$: Beacon transmission, Sleeping, and Listening. Nodes consume a specific amount of energy ε_v in each state v.

In practice, since battery voltage drops over time, battery capacity is often measured with normalized voltage. Therefore, energy is represented in the units of A·s. In sleeping and listening states, the energy consumed during a time unit, T_u, are $\varepsilon_\mathrm{sl} = I_\mathrm{sl} T_\mathrm{u}$, and $\varepsilon_\mathrm{li} = I_\mathrm{li} T_\mathrm{u}$, respectively, where I_sl and I_li are the measured current drawn from the battery in the sleep and listening modes, respectively.

The power consumption when the node is transmitting beacon packets, ε_b, depends on the beacon transmission process shown in Fig. 10. The node transmits beacons only in a portion of time, and the portion, ω_b, should be obtained first to determine ε_b. For a node within ring r, ω_b is expressed as

$$\omega_\mathrm{b}(r) = \frac{T_\mathrm{tx}}{\left(\frac{T_\mathrm{ibo}^\mathrm{max}}{2} + \frac{T_\mathrm{cbo}^\mathrm{max} p_\mathrm{busy}(r)}{2(1-p_\mathrm{busy}(r))^2} + T_\mathrm{tx} + T_\mathrm{to}\right)}, \tag{69}$$

where $p_\mathrm{busy}(r)$ is the probability of sensing the channel busy, and is derived as follows.

First, the region within the transmission range of location $\boldsymbol{x}$, $\mathbb{C}(\boldsymbol{x})$, is divided into small areas according to the polar coordinates centered at the sink. Thus, in the small area $(r : r + \Delta r, \theta : \theta + \Delta\theta)$, denote $p_\mathrm{ex}(r)$ as the probability that there exists a node in this area, and $\phi_\mathrm{b}(r)$ as the probability that at any time a node in this area, if it exists, is transmitting a beacon packet. Then $p_\mathrm{busy}(r)$ is given by

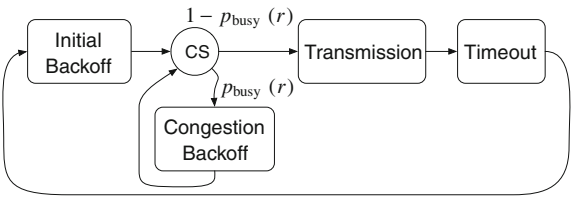


Fig. 10 The process of transmitting beacon packets

$$p_{\text{busy}}(r) = 1 - \prod_{\boldsymbol{y}=(r',\theta')\in\mathbb{C}(\boldsymbol{x})} \Big(1 - p_{\text{ex}}(r')\phi_{\text{b}}(r')\Big), \tag{70}$$

where $p_{\text{ex}}(r)$ is given by

$$p_{\text{ex}}(r) = \rho \Delta r \Delta\theta r, \tag{71}$$

where ρ is the node density. The probability that a node in this area is transmitting a beacon packet, $\phi_{\text{b}}(r)$, is given by $\phi_{\text{b}}(r) = \pi_{\text{b}}(r)\omega_{\text{b}}(r)$, where $\pi_{\text{b}}(r)$ is the total probability that the node is in one of the beacon transmission states in the DTMC $\{X_n\}$, and is given by adding the probabilities in the equilibrium state probability vector, $\boldsymbol{\pi}(r)$, corresponding to the beacon transmission states. Therefore, according to (70), for nodes located at $\boldsymbol{x}$ in ring r, the portion of time in which they transmit beacon messages, $\omega_{\text{b}}(r)$, depends on its values for other nodes in its neighborhood, $\mathbb{C}(\boldsymbol{x})$. An iterative procedure is used for all r's to calculate $\omega_{\text{b}}(r)$ at the end of Sect. 5.5.

Then, $\varepsilon_{\text{b}}(r)$ is obtained by

$$\varepsilon_{\text{b}}(r) = \Big(I_{\text{li}}(1 - \omega_{\text{b}}(r)) + I_{\text{tx}}\omega_{\text{b}}(r)\Big)T_{\text{u}}, \tag{72}$$

where I_{li} and I_{tx} is the measured current when the node is listening and transmitting, respectively.

Finally, for this case study, we assume that the data processing time is far shorter than a time unit T_{u}.

Since data processing is conducted when packets are generated or received, a fixed amount of energy, ε_{p}, is added to the energy consumption in the first state of each $\{C_n\}$.

5.5.2 Communication and Data Processing Energy Consumption

The other parameters in $\{X_n\}$, i.e., transition probability matrices and traffic rates in $\{I_n\}$ and $\{C_n\}$, are obtained according to the discussion in Sect 3.5. Then, the equilibrium state probability vector, $\boldsymbol{\pi}(r)$, for the DTMC $\{X_n\}$ is obtained for each node $\boldsymbol{x}$. It should be noted that while we solve $\omega_{\text{b}}(r)$, it is assumed that $\omega_{\text{b}}(r')$ for all nodes $\boldsymbol{y}$ in range are known. This dependency is solved in an iterative manner. First, initial guesses of $\omega_{\text{b}}(r)$ for all rings are set to all 0's in our evaluation. Then, updated values of $\omega_{\text{b}}(r)$ are calculated. The iteration terminates when the difference between two consecutive iterations is negligible for each ring. Then, the energy consumption during a beacon time unit, $\varepsilon_{\text{b}}(r)$, is obtained according to (72). Finally, the communication and data processing energy consumption distribution for any single node is calculated according to (57).

5.5.3 Total Energy Consumption and Lifetime Distribution

Finally, additional energy consumed by sensing activities are considered. The distribution of total energy consumption of the node is then obtained.

With the energy consumption distribution for nodes in each ring known, the lifetime distribution for nodes in each ring, $L(r, C)$, is directly obtained by (66). Then, the distribution of the network lifetime, and its Normal distribution approximation are obtained.

5.6 Experiments

We analyze the accuracy of the models by comparisons to testbed experiments and simulations. Similar parameters are used for the experiments as before.

5.6.1 Validation of the Single-node Energy Analysis

We first analyze the energy consumption distribution in (57). The energy consumption distributions during $T = 60$ s for two nodes with distances of 2.6 and 3.5 m to the sink are measured. The *cdf*s of the measured energy consumption are shown in Fig. 11 with the analytical model results. It can be observed that the error of the analytical *cdf* is less than 5 % compared to the empirical measurements for each node. It is also observable that the *cdf*s for the node at $r = 3.5$ m exhibit a steep increase at the energy level of 0.6 A·s. This is because there is a high probability that the node consumes exactly 0.6 A·s energy, which corresponds to the case where nodes are performing their normal duty cycle operations. The same network topology is also simulated using TOSSIM, which also agree with the results.

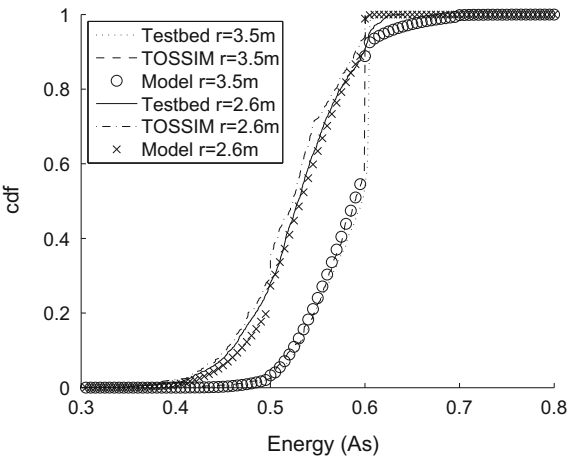


Fig. 11 *cdf* of the energy consumption during 1 min. Testbed experiments, simulation, and analytical results are shown

5.6.2 Validation of the Normal Distribution Approximation

The asymptotic Normal distribution approximation of energy consumption for large T is validated using simulations [70]. Each topology is simulated for 10 days, and 100 different topologies are generated. In addition, additional energy consumptions for sensing and data processing are added to simulate a fully operational WSN application. The *cdf* of energy consumption for node at $r = 27$ m for $T = 2$ and 30 min are shown in Fig. 12a and b. The *cdf* of the asymptotic Normal distributions in Theorem 3 are also shown.

It can be observed in Fig. 12a and b that as the duration increases, the energy consumption distribution converges to the asymptotic Normal distribution.

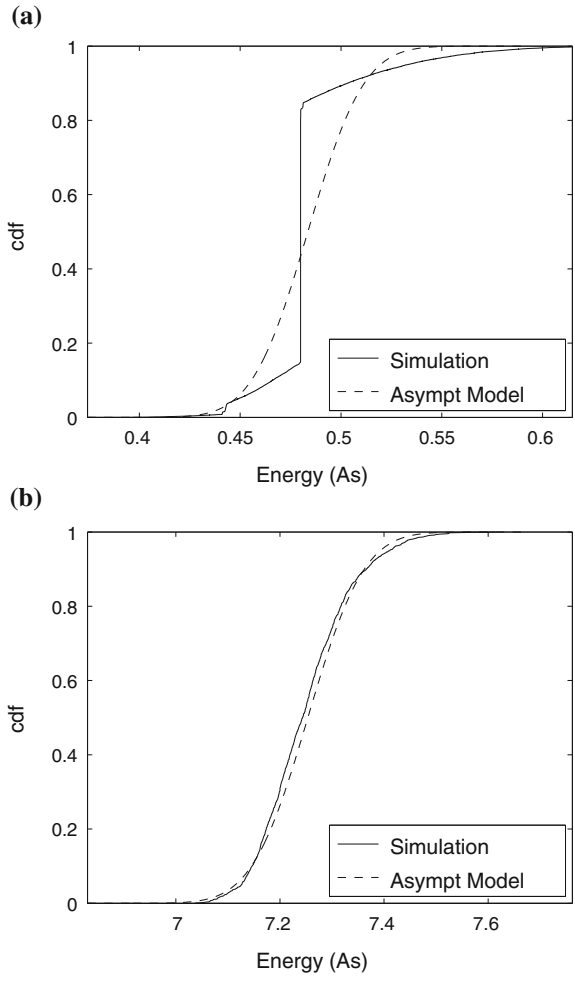


Fig. 12 *cdf* of the energy consumption during longer periods. As the duration increases, the energy consumption approaches the asymptotic normal distribution (**a**) T = 2 min (**b**)T = 30 min

6 Conclusion

In this chapter, the stochastic analysis of end-to-end communication delay, event detection delay, energy consumption, and lifetime in WSNs is discussed. A Markov process based on the birth-death problem is used to model the transmission process in a multi-hop network. Accordingly, the effects of wireless channel errors and queuing delays on communication delay, energy consumption, and lifetime can be captured. The model is validated by extensive testbed experiments through several network configurations and parameters. The results show that the MArkov-based approach accurately models the distribution of the end-to-end delay and captures the heterogeneous effects of multi-hop WSNs.

An analytical model is also described to model the event detection in WSNs. In the framework, a spatio-temporal fluid model is utilized to obtain the distribution of the event detection delay. The average delay and soft delay bounds are then obtained. To reduce the calculation complexity, a simplified model is also derived, motivated by the fact that the queue build up in WSNs is negligible. Testbed experiments and simulations are used to validate the accuracy of both approaches.

Finally, the probabilistic analysis of the energy consumption is provided. Energy consumption for communication, data processing, and sensing are all captured by the analytical framework. The energy consumption distribution for each node is derived. It is shown that, when the time duration is long, the energy consumption converges to a Normal distribution, and the mean and variance of such distribution are also provided. With the help of energy consumption distribution, the lifetime distributions for each node and the entire network are derived. The described model is validated by both testbed experiments and simulations. The results show that the distribution of the energy consumption can be accurately modeled.

References

1. T. Abdelzaher, S. Prabh, R. Kiran, On real-time capacity limits of multihop wireless sensor networks, in *Proceedings of IEEE RTSS 2004*, pp. 359–370 (2004). doi:10.1109/REAL.2004.37
2. K. Akkaya, M. Younis, A survey on routing protocols for wireless sensor networks. Ad Hoc Netw. **3**(3), 325–349 (2005)
3. I.F. Akyildiz, T. Melodia, K.R. Chowdhury, A survey on wireless multimedia sensor networks. Comput. Netw. **51**(4), 921–960 (2007). doi:http://dx.doi.org/10.1016/j.comnet.2006.10.002
4. I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, Wireless sensor networks: a survey. Comput. Netw. J. (Elsevier) **38**(4), 393–422 (2002). doi:http://dx.doi.org/10.1016/S1389-1286(01),00302-4
5. G. Armitage, *Quality of Service in IP Networks: Foundations for a Multi-Service Internet* (Macmillan Publishing Co., Inc. Indianapolis, 2000)
6. G. Bianchi, Performance analysis of the IEEE 802.11 distributed coordination function. IEEE J. Sel. Areas Commun. **18**(3), 535–547 (2000). doi:10.1109/49.840210

7. N. Bisnik, A. Abouzeid, Queuing network models for delay analysis of multihop wireless ad hoc networks, in *IWCMC 2006: Proceedings of the 2006 International Conference on Wireless Communications and Mobile Computing*, pp. 773–778. Vancouver, British Columbia, Canada (2006). doi:10.1145/1143549.1143704
8. M. Buettner, G.V. Yee, E. Anderson, R. Han, X-MAC: a short preamble MAC protocol for duty-cycled wireless sensor networks, in *Proceedings of ACM SenSys 2006*. Boulder, CO (2006)
9. A. Burchard, J. Liebeherr, S. Patek, A min-plus calculus for end-to-end statistical service guarantees. IEEE Trans. Inf. Theory **52**(9), 4105–4114 (2006). doi:10.1109/TIT.2006.880019
10. Q. Cao, T. Yan, J. Stankovic, T. Abdelzaher, Analysis of target detection performance for wireless sensor networks, in *Proceedins of DCOSS 2005*, pp. 276–292. Marina del Rey, CA (2005)
11. S. Chakrabarti, A. Mishra, QoS issues in ad hoc wireless networks. IEEE Commun. Mag. **39**(2), 142–148 (2002)
12. D. Chen, P. Varshney, QoS support in wireless sensor networks: a survey, in *International Conference on Wireless Networks*, pp. 227–233. Citeseer (2004)
13. C. Chiasserini, R. Gaeta, M. Garetto, M. Gribaudo, D. Manini, M. Sereno, Fluid models for large-scale wireless sensor networks. Perform. Eval. **64**(7–8), 715–736 (2007)
14. E. Crawley, R. Nair, B. Rajagopalan, H. Sandick, A Framework for QoS-based Routing in the Internet. RFC2386, August (1998)
15. R. Cruz, A calculus for network delay. I. Network elements in isolation. IEEE Trans. Inf. Theory **37**(1), 114–131 (1991). doi:10.1109/18.61109
16. W. Dargie, X. Chao, M. Denko, Modelling the energy cost of a fully operational wireless sensor network. Telecommun. Syst. **44**(1–2), 3–15 (2010)
17. O. Dousse, C. Tavoularis, P. Thiran, Delay of intrusion detection in wireless sensor networks, in *Proceedings of ACM MobiHoc 2006*, p. 165. Florence, Italy (2006)
18. E. Duarte-Melo, M. Liu, Analysis of energy consumption and lifetime of heterogeneous wireless sensor networks, in *Proceedings of IEEE GLOBECOM 2002*. Taipei, Taiwan (2002)
19. E. Duarte-Melo, M. Liu, A. Misra, A modeling framework for computing lifetime and information capacity in wireless sensor networks, in *WiOpt 2004*. Cambridge, UK (2004)
20. E.J. Duarte-melo, M. Liu, Data-gathering wireless sensor networks: Organization and capacity. Comput. Netw. **43**, 519–537 (2003)
21. E. Felemban, C.G. Lee, E. Ekici, R. Boder, S. Vural, Probabilistic QoS guarantee in reliability and timeliness domains in wireless sensor networks, in *Proceedings of IEEE INFOCOM 2005*, vol. 4, pp. 2646–2657. Miami, FL (2005). doi:10.1109/INFCOM.2005.1498548
22. M. Fidler, An end-to-end probabilistic network calculus with moment generating functions, in *Proceedings of IEEE IWQoS*, pp. 261–270. New Haven, CT (2006). doi:10.1109/IWQOS.2006.250477
23. M. Franceschetti, O. Dousse, D. Tse, P. Thiran, Closing the gap in the capacity of wireless networks via percolation theory. IEEE Trans. Inf. Theory **53**(3), 1009–1018 (2007)
24. R.G. Gallager, *Discrete Stochastic Processes* (Kluwer Academic Publishers, New York, 1995)
25. L. Galluccio, S. Palazzo, End-to-End delay and network lifetime analysis in a wireless sensor network performing data aggregation, in *Proceedings of IEEE GLOBECOM 2009*, pp. 146–151. Honolulu, HI (2009)
26. K. Gopalan, T.C. Chiueh, Y.J. Lin, Probabilistic delay guarantees using delay distribution measurement, in *Proceedings of ACM MULTIMEDIA 2004*, pp. 900–907. New York, NY (2004). doi: http://doi.acm.org/10.1145/1027527.1027734
27. M. Gribaudo, D. Manini, A. Nordio, C. Nordio, Analysis of IEEE 802.15. 4 sensor networks for event detection, in *Proceedings of IEEE Globecom 2009*, pp. 152–157. Honolulu, Hawaii (2009)
28. G.R. Gupta, N.B. Shroff, Delay analysis for multi-hop wireless networks, in *Proceedings of IEEE INFOCOM 2009*, pp. 412–421. Rio de Janeiro, Brazil (2009)
29. P. Gupta, P.R. Kumar, The capacity of wireless networks. IEEE Trans. Inf. Theory IT-**46**(2), 388–404 (2000)

30. J. Haapola, Z. Shelby, C. Pomalaza-Raez, P. Mahonen, Cross-layer energy analysis of multihop wireless sensor networks, in *Proceedings of IEEE EWSN 2005* (2005)
31. T. Issariyakul, E. Hossain, Analysis of end-to-end performance in a multi-hop wireless network for different hop-level ARQ policies, in *Proceedings of IEEE GLOBECOM 2004*, vol. 5, pp. 3022–3026. Dallas, TX (2004). doi:10.1109/GLOCOM.2004.1378907
32. D. Jung, T. Teixeira, A. Savvides, Sensor node lifetime analysis: models and tools. ACM Trans. Sens. Netw. **5**(1), 1–33 (2009). doi:10.1145/1464420.1464423
33. J. Kim, X. Lin, N. Shroff, Optimal anycast technique for delay-sensitive energy-constrained asynchronous wireless sensor networks, in *Proceedings of IEEE INFOCOM 2009*, pp. 412–421. Rio de Janeiro, Brazil (2009)
34. A. Koubaa, M. Alves, E. Tovar, Modeling and worst-case dimensioning of cluster-tree wireless sensor networks, in *Proceedings of IEEE RTSS 2006*, pp. 412–421. Rio de Janeiro, Brazil (2006). doi:10.1109/RTSS.2006.29
35. S. Kulkarni, A. Iyer, C. Rosenberg, An address-light, integrated mac and routing protocol for wireless sensor networks. IEEE/ACM Trans. Netw **14**(4), 793–806 (2006). doi:10.1109/TNET.2006.880163
36. V.G. Kulkarni, Fluid models for single buffer systems. Front. Queue. Models Appl. Sci. Eng., pp. 321–338 (1997)
37. S. Kumar, A. Arora, T. Lai, On the lifetime analysis of always-on wireless sensor network applications, in *Proceedings of IEEE MASS 2005*. Washington, DC (2005)
38. J. Lehoczky, Real-time queueing network theory, in *Proceedings of IEEE RTSS 1997*, pp. 58–67. San Francisco, CA (1997). doi:10.1109/REAL.1997.641269
39. P. Levis, N. Lee, M. Welsh, D. Culler, TOSSIM: accurate and scalable simulation of entire tinyos applications, in *Proceedings of ACM SenSys 2003*. Los Angeles, CA (2003)
40. H. Li, P. Shenoy, K. Ramamritham, Scheduling messages with deadlines in multi-hop real-time sensor networks, in *Proceedings of IEEE RTAS 2005*, pp. 415–425. San Francisco, CA (2005). doi:10.1109/RTAS.2005.48
41. S. Liu, K.W. Fan, P. Sinha, CMAC: An energy efficient mac layer protocol using convergent packet forwarding for wireless sensor networks, in *Proceedings of SECON 2007*, pp. 11–20. San Diego, CA (2007). doi:10.1109/SAHCN.2007.4292813
42. Y. Liu, F. Lo Presti, V. Misra, D. Towsley, Y. Gum Fluid models and solutions for large-scale IP networks, in *Proceedings of ACM SIGMETRICS 2003*, pp. 91–101. San Diego, CA (2003)
43. D. Marco, E.J. Duarte-Melo, M. Liu, D.L. Neuhoff, On the many-to-one transport capacity of a dense wireless sensor network and the compressibility of its data, in *Proceedings of IPSN'03*. Palo Alto, CA (2003)
44. W. Marcus, An architecture for QoS analysis and experimentation. IEEE/ACM Trans. Netw. (TON) **4**(4), 603 (1996)
45. D. McDysan, *QoS and Traffic Management in IP and ATM Networks* (McGraw-Hill, New York, 1999)
46. T. Melodia, M.C. Vuran, D. Pompili, The state of the art in cross-layer design for wireless sensor networks, in *Wireless Systems and Network Architectures in Next Generation Internet*, pp. 78–92 (2006)
47. P. Mohapatra, J. Li, C. Gui, QoS in mobile ad hoc networks. IEEE Wireless Commun. **10**(3), 44–53 (2003)
48. A. Muhammad Mahtab, B. Olivier, M. Daniel, A. Thomas, S. Olivier, A hybrid model for accurate energy analysis of WSN nodes. EURASIP J. Embed. Syst. **2011** (2011)
49. Y. Nazarathy, G. Weiss, The asymptotic variance rate of the output process of finite capacity birth-death queues. Queue. Syst. **59**(2), 135–156 (2008). doi:10.1007/s11134-008-9079-4
50. R. Nelson, *Probability, Stochastic Processes, and Queueing Theory: The Mathematics of Computer Performance Modeling* (Springer, New York, 1995)
51. M. Neuts, J. Guo, M. Zukerman, H.L. Vu, The waiting time distribution for a TDMA model with a finite buffer and state-dependent service. IEEE Trans. Commun. **53**(9), 1522–1533 (2005). doi:10.1109/TCOMM.2005.855014

52. M.F. Neuts, Matrix-Geometric Solutions in Stochastic Models: an Algorithmic Approach (Dover, New York, 1981)
53. M. Noori, M. Ardakani, Lifetime analysis of random event-driven clustered wireless sensor networks. IEEE Trans. Mob. Comput. **10**(10), 1448–1458 (2010)
54. R. Oliver, G. Fohler, Probabilistic estimation of end-to-end path latency in wireless sensor networks, in *Proceedings of IEEE MASS 2009*, pp. 423–431. Macau, China (2009). doi:10.1109/MOBHOC.2009.5336970
55. W. Pak, J.G. Choi, S. Bahk, Tier based anycast to achieve maximum lifetime by duty cycle control in wireless sensor networks, in *Wireless Communications and Mobile Computing Conference, 2008*. IWCMC 2008. International, pp. 123–128 (2008). doi:10.1109/IWCMC.2008.22
56. P. Park, P. Di Marco, P. Soldati, C. Fischione, K. Johansson, A generalized markov chain model for effective analysis of slotted IEEE 802.15.4, in *Proceedings of IEEE MASS 2009*, pp. 130–139. Macau, China (2009). doi:10.1109/MOBHOC.2009.5337007
57. J. Polastre, J. Hill, D. Culler, Versatile low power media access for wireless sensor networks, in *Proceedings of ACM SenSys 2004*. Baltimore, MA (2004). doi: http://doi.acm.org/10.1145/1031495.1031508
58. S. Pollin, M. Ergen, S. Ergen, B. Bougard, F. Catthoor, A. Bahai, P. Varaiya, Performance analysis of slotted carrier sense ieee 802.15.4 acknowledged uplink transmissions, in *Proceedings of IEEE WCNC 2008*, pp. 1559–1564. Las Vegas, NV (2008). doi:10.1109/WCNC.2008.279
59. T. Sakurai, H. Vu, MAC access delay of IEEE 802.11 DCF. IEEE Trans. Wireless Commun. **6**(5), 1702–1710 (2007). doi:10.1109/TWC.2007.360372
60. J. Schmitt, F. Zdarsky, L. Thiele, A comprehensive worst-case calculus for wireless sensor networks with in-network processing, in *RTSS 2007*, pp. 193–202 (2007). doi:10.1109/RTSS.2007.17
61. O. Tickoo, B. Sikdar, Modeling queueing and channel access delay in unsaturated IEEE 802.11 random access MAC based wireless networks. IEEE/ACM Trans. Netw. **16**(4), 878–891 (2008). doi:http://dx.doi.org/10.1109/TNET.2007.904010
62. MICA2 sensor node. http://www.xbow.com
63. TelosB sensor node. http://www.xbow.com
64. S. Toumpis, L. Tassiulas, Packetostatics: deployment of massively dense sensor networks as an electrostatics problem, in *Proceedings of IEEE INFOCOM 2005*. Miami, FL (2005)
65. L. Van Hoesel, T. Nieberg, J. Wu, P. Havinga, Prolonging the lifetime of wireless sensor networks by cross-layer interaction, in *IEEE Wireless Communications*, p. 79 (2004)
66. M.C. Vuran, I.F. Akyildiz, XLP: a cross layer protocol for efficient communication in wireless sensor networks. IEEE Trans. Mob. Comput. **9**(11), 1578–1591 (2010)
67. Q. Wang, M. Hempstead, W. Yang, A realistic power consumption model for wireless sensor network devices, in *Proceedings of IEEE SECON 2006*, pp. 286–295. Reston, VA (2006)
68. Y. Wang, M.C. Vuran, S. Goddard, Cross-layer analysis of the end-to-end delay distribution in wireless sensor networks, in *Proceedings of IEEE RTSS 2009*. Washington, DC (2009)
69. Y. Wang, M.C. Vuran, S. Goddard, Stochastic analysis of energy consumption in wireless sensor networks, in *Proceedings of IEEE SECON 2010*. Boston, MA (2010)
70. Y. Wang, M.C. Vuran, S. Goddard: Analysis of event detection delay in wireless sensor networks, in *Proceedings of IEEE INFOCOM 2011*. Shanghai, China (2011)
71. Y. Wang, M.C. Vuran, S. Goddard, Cross-layer analysis of the end-to-end delay distribution in wireless sensor networks. IEEE Trans. Netw. **20**(1), 305–318 (2012)
72. M. Xie, M. Haenggi, Towards an end-to-end delay analysis of wireless multihop networks. Ad Hoc Netw. **7**(5), 849–861 (2009). doi:10.1016/j.adhoc.2008.04.010
73. Y. Xue, M.C. Vuran, B. Ramamurthy, Cost-efficiency of anycast-based forwarding in duty-cycled wsns with lossy channel, in *Proceedings of IEEE SECON 2010*. Boston, MA (2010)
74. W. Ye, J. Heidemann, D. Estrin, Medium access control with coordinated adaptive sleeping for wireless sensor networks. IEEE/ACM Trans. Netw. **12**(3), 493–506 (2004). doi: http://dx.doi.org/10.1109/TNET.2004.828953

75. S.N. Yeung, J. Lehoczky, End-to-end delay analysis for real-time networks, in *Proceedings of IEEE RTSS 2001*, pp. 299–309. London, UK (2001)
76. M. Zúniga, B. Krishnamachari, An analysis of unreliability and asymmetry in low-power wireless links. ACM Trans. Sens. Netw. **3**(2) (2007)

Part II
Barrier and Spatiotemporal Coverage

Chapter 3
Barrier Coverage: Foundations and Design

Anwar Saipulla, Jun-Hong Cui, Xinwen Fu, Benyuan Liu and Jie Wang

Abstract The coverage of a wireless sensor network (WSN) characterizes the quality of surveillance that the WSN can provide. A deep understanding of the coverage is of great importance for the deployment, design, and planning of wireless sensor networks. Barrier coverage measures the capability of a wireless sensor network to detect intruders that attempt to cross the deployed region. The goal is to prevent intruders from sneaking through the network undetected. It is a critical issue for many military and homeland security applications. In this chapter we provide a comprehensive survey on the barrier coverage of wireless sensor networks. The main topics include the critical conditions and construction of barrier coverage in a 2-dimensional WSN, the barrier coverage under a line-based sensor deployment scheme, the effect of sensor mobility on barrier coverage, and the barrier coverage for a 3-dimensional underwater sensor network. For each topic we discuss the challenges, fundamental limits, and the solution for the construction of sensor barriers.

A. Saipulla (✉) · X. Fu · B. Liu · J. Wang
University of Massachusetts Lowell, Lowell, MA 01854, USA
e-mail: asaipull@cs.uml.edu

X. Fu
e-mail: xinwenfu@cs.uml.edu

B. Liu
e-mail: bliu@cs.uml.edu

J. Wang
e-mail: wang@cs.uml.edu

J-H. Cui
University of Connecticut, Storrs, CT 06269, USA

H. M. Ammari (ed.), *The Art of Wireless Sensor Networks*,
Signals and Communication Technology, DOI: 10.1007/978-3-642-40066-7_3,

1 Introduction

A wireless sensor network is typically used to monitor its deployed region. The coverage of the network is a very important performance metric, characterizing the quality of surveillance that the WSN can provide, for example, how well a region of interest is monitored by sensors; how effective a sensor network is in detecting intruders, etc. Without desired coverage, a sensor network cannot fulfill its mission. A good coverage provides necessary basis for high-level functions such as intrusion detection, target localization, classification, and tracking.

Coverage requirements vary with different application scenarios. In some applications, the goal is to efficiently deploy sensors so as to cover a set of designated target points, and the problem is often referred to *point coverage*. *Area coverage* concerns the fraction of area covered by one or more sensors in the deployed region. The goal is to efficiently cover the whole area of the deployed region. *Barrier coverage* characterizes the capability of a wireless sensor network to detect intruders that attempt to cross the deployed region. The goal is to prevent intruders from sneaking through the network undetected. *Sweeping coverage* represents the dynamic coverage of a mobile sensor network as mobile sensors move around the field to cover initially uncovered areas and detect intruders.

Barrier coverage of wireless sensor networks provides sensor barriers guarding boundaries of critical infrastructures or assets, such as country borders, coastal lines, and boundaries of battlefields. It is a critical issue for many military and homeland security applications. Due to its unique requirement, barrier coverage exhibits different characteristics and calls for different design considerations than other coverage measures. A sensor barrier is formed by a connected sensor cluster across the entire deployed region, acting as a "trip wire" to detect any crossing intruders. This is illustrated in Fig. 1.

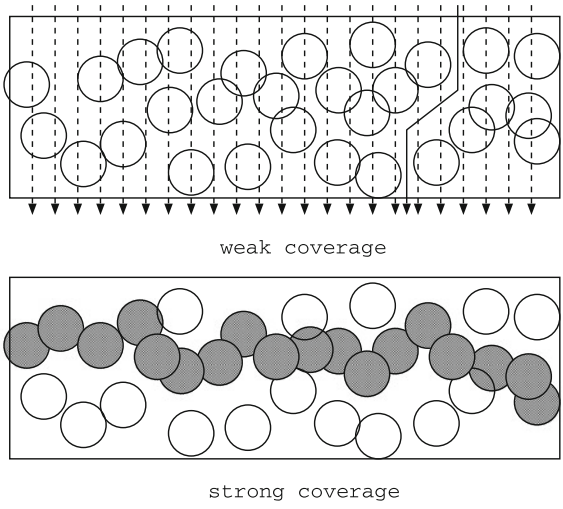


Fig. 1 Weak barrier coverage guarantees to detect intruders on congruent crossing paths. Strong barrier coverage guarantees to detect intruders for arbitrary crossing paths. A sensor barrier acts as a "trip wire" to detect crossing intruders

Providing barrier coverage in a wireless sensor network faces a number of challenges. The barrier coverage of a wireless sensor network depends on many factors, for example, sensor deployment strategies, sensing characteristics, sensor collaboration scheme, node mobility, etc. It is important to obtain a fundamental understanding of how these factors affect the barrier coverage. Furthermore, the design of barrier construction and sensor mobility schemes is a challenging task due to the sheer scale of the network and the non-local nature of the barrier coverage problem.

In this chapter, we provide a survey of recent research activities and results in the area of sensor barrier coverage. In particular, we will focus on the following research thrusts:

- **Critical conditions and barrier construction.** In some applications sensors may be manually deployed so barrier coverage can be achieved using a minimum number of sensors by aligning them in straight lines crossing the network region. In other applications, sensors may have to be deployed randomly. It is important to establish the theoretical foundation for the deployment, design, and performance in connection to barrier coverage.
  We will first present the critical condition for barrier coverage in a two-dimensional rectangular region where sensors are placed uniformly at random [1]. In particular, the existence and strength of the barrier coverage depends on the width-to-length ratio of the rectangle area. If the width of the rectangular area is asymptotically smaller than the logarithm of the length, the probability that there exist crossing paths that are not covered by sensors approaches 1 and the network has no barrier coverage. On the other hand, if the width is asymptotically larger than the logarithm of the length, the barrier coverage starts to appear when the node density reaches a certain value. There exist multiple disjoint sensor barriers across the entire length of the region such that intruders cannot cross undetected. The analytical characterizations of the barrier coverage can be used to determine the number of barriers for a given network deployment, and the minimum number of sensors needed to satisfy a given barrier coverage requirement.
  Building upon the theoretical foundations established for the barrier coverage problem, we will present an efficient distributed algorithm to construct disjoint barriers in a long rectangle area [1]. Specifically, the original network is divided into small segments interleaved by thin vertical strips. Each segment and vertical strip independently computes the barriers in its own area. We construct horizontal barriers in each segment connected by vertical barriers in neighboring vertical strips. Continuous barrier coverage for the whole region is thus guaranteed. By dividing the network into small segments and letting each segment conduct communication and computation independently, our algorithm can significantly reduce delay, communication overhead, and computation costs compared to the early centralized approach. Moreover, since each segment computes the barriers between vertical barriers at the two ends, a larger number of local barriers will be obtained, resulting in strengthened local barrier coverage for each segment.
- **Sensor Deployment.** The choice of sensor deployment strategies has direct impact on barrier coverage. While manually placing sensors side by side along straight

lines across the region yields the most efficient barrier coverage, it is infeasible to be implemented in many application scenarios such as hostile environments and hard-to-reach areas. A widely used random sensor deployment model assumes that sensors are distributed in a vast region uniformly at random and is approximated by a Poisson point process. For this model, it has been shown that certain conditions have to be met and the deployment can be inefficient for achieving barrier coverage [1]. Moreover, while the model may be appropriate for some deployment strategies, the uniform distribution of sensors does not capture the sensor location distribution under many other deployment strategies.

When deploying sensors to monitor boundaries of battlefields or country borders with complex terrains, a viable approach is to drop sensors from aircrafts along predetermined routes [2, 3]. When dispersed from an aircraft, sensors will be scattered and centered around the deployed line with some random offsets due to mechanical inaccuracy, wind, terrain characteristics, and other factors. The line-based sensor deployment strategy is of particular interest to military and homeland security applications.

We will first present the assumptions and the corresponding analytical results for the probability of barrier coverage under the line-based sensor deployment strategy. We will then validate the analysis results via simulations, and compare the barrier coverage under line-based sensor deployment strategy with those under two-dimensional and one-dimensional uniformly random sensor deployment models [4].

- **Sensor Mobility.** In a sensor network, barrier gaps may occur at deployment if sensors are deployed randomly, or in an already formed barrier if some sensors used to form the barrier run out of power or start malfunctioning, resulting in a degraded barrier coverage. Recent technology advances have made it possible to deploy mobile sensors in practical applications [5, 6]. Sensor mobility can be exploited to improve the barrier coverage in a wireless sensor network. After the initial deployment, mobile sensors can move to connect with other sensors so as to form new barriers. However, simply moving sensors to form a large local cluster does not necessarily yield a global barrier. This global nature of barrier coverage makes it a challenging task to devise effective sensor mobility schemes. Moreover, a good sensor mobility scheme should efficiently improve barrier coverage under the constraints of the number of mobile sensors and their moving range.

  We will first explore the fundamental limits of sensor mobility on barrier coverage and present a sensor mobility scheme that constructs the maximum number of barriers with minimum sensor moving distance [7]. We will then present an efficient algorithm to compute the existence of barrier coverage with sensors of limited mobility, and examine the effects of the number of mobile sensors and their moving ranges on the barrier coverage improvement. Finally, we will present a two-phase algorithm that first find barrier gaps and then relocate mobile sensors to fill the gaps and form new barriers, while at the same time, the energy consumptions among the mobile sensors are balanced to prolong the network lifetime [8].
- **Underwater Sensor Barriers.** Anti-submarine warfare(ASW) is a critical challenge for maintaining a fleet presence in hostile areas. Technology advancement

has allowed submarines to evade standard sonar detection. A viable alternative is to place magnetic or acoustic sensors in close proximity to possible underwater pathways of submarines. This approach may require deploying large-scale underwater sensor networks to form 3-dimensional barriers.
We will present new results on sensor barriers for 3-dimensional sensor networks [9]. We will first show that when sensor locations follow a Poisson Point Process then sensor barriers in a large 3-dimensional fixed emplacement sensor field are unlikely to exist. We use the notion of 3-dimensional stealth distance to measure how far a submarine can travel in a sensor network without being detected. We will then devise an energy-conserving scheme to construct 3-dimensional barriers using mobile sensors. We focus on developing an energy efficient matching of mobile sensors to cover grid points using distributed auction algorithms. Specifically, we try to minimize the maximum travel distance between any sensor and its assigned grid point. Through simulation we show in comparison to the optimal solution, the distributed auction-based approach offers reduced computation time and similar maximum travel distance. This provides a promising new approach to constructing barriers in 3-dimensional sensor networks.

The remainder of this chapter is organized as follows. Section 2 provides a review of the research on barrier coverage of wireless sensor networks. In Sect. 3, we present the critical conditions for the barrier coverage and algorithms for efficient barrier construction. Section 4 presents the barrier coverage under line-based sensor deployment strategy. Section 5 investigates the fundamental limits and efficient mobility schemes of exploiting sensor mobility to improve barrier coverage. Section 6 presents the barrier coverage construction in underwater sensor networks. Finally, Sect. 7 summaries this chapter.

2 Related Work

The coverage, deployment, and target tracking problems in wireless sensor networks have been extensively studied in the past decade, e.g., [10–16], to name just a few. The joint coverage and connectivity has also been investigated recently [17–19].

The notion of barrier coverage was first introduced in the context of robotics sensors [20]. The goal of barrier coverage is to detect intruders that attempt to cross from one side of a region to the opposite side. A number of different barrier coverage measures and the related issues have been studied.

In [21], Liu and Towsley studied the barrier coverage of two-dimensional plane and two-dimensional strip sensor networks using percolation theory results. The barrier coverage of a two-dimensional plane network is related to the existence of a giant sensor cluster that percolates the network. For a two-dimensional trip network of finite width, it is proved that there always exist crossing paths along which an intruder can cross the strip undetected. Furthermore, the probability that an intruder is detected when crossing a strip is characterized. In [1], the critical conditions of the

existence and strength of the barrier coverage are established. Moreover, efficient algorithm has been devised to construct sensor barriers with low communication and computation overhead. Most of the studies on barrier coverage assume that the sensor locations follow a Poisson point process where sensors are distributed in a large area uniformly at random. Saipulla et al. considered an application scenario where sensors are dropped along certain lines with random offsets, and established analytical results for the barrier coverage probability in [4].

In a mobile sensor network, sensor mobility can be exploited for autonomous barrier coverage construction and improvement. Saipulla et al. studied how to relocate sensors with limited mobility to improve barrier coverage after the initial deployment [7]. They derived the fundamental limits of sensor mobility on barrier coverage, and devised algorithms that can check the existence of barrier coverage under sensor mobility and construct the maximum number of barriers with minimum sensor moving distance. In [22], Shen et al. proposed a virtual force-based heuristic algorithm to relocate mobile sensors to form barriers. To address the deployment cost problem, He et al. designed a periodic monitoring scheduling algorithm in which each point along the barrier line is monitored periodically by mobile sensors [23], and proposed a coordinated sensor patrolling algorithm to further improve the barrier coverage. In [24], Kong et al. studied the mobile barrier coverage surrounding dynamic objects.

In [25], Kumar et al. introduced a notion of weak barrier coverage and derived the critical conditions to achieve weak barrier coverage in a randomly deployment sensor network. While strong barrier coverage guarantees the detection of intruders no matter what crossing paths they take, the weak barrier coverage guarantees to detect intruders moving along congruent paths. Figure 1 illustrates the difference between the two different barrier coverage measures. In the top figure, the network has weak barrier coverage for all orthogonal crossing paths (dashed paths). However, there is an uncovered path (solid path) through the region. The bottom figure shows an example of strong barrier coverage where no intruders can cross the region undetected, no matter how they choose their crossing paths. The barrier is highlighted using shaded sensing areas. In the remainder of the chapter we will focus on the strong barrier coverage and refer to it as barrier coverage for conciseness.

Many other aspects of barrier coverage have also been investigated. In [26, 27], Yang and Qiao studied the effects of sensor collaboration and multi-round deployment on barrier coverage. Barrier coverage of camera-based sensor network have been studied in [28, 29]. In [30], Chen et al. introduced the notion of local barrier coverage and devised a localized algorithm that guarantees the detection of intruders whose trajectory is confined to a slice of the belt region of deployment. It does not protect the network against intruders that can move beyond the range of the thin slices. In [31], the measurement and the quality of barrier coverage in wireless sensor networks are investigated. The effect of directional sensors on barrier coverage is considered in [32], where an integer linear programming formulation is used to provide optimal solutions, and centralized algorithms and a distributed algorithm are proposed to solve the problem.

3 Critical Conditions and Barrier Construction

3.1 Network Model

Sensors are assumed to be deployed in a *two-dimensional strip area* of size $A_{\text{2-dim strip}} = [0, n] \times [0, w(n)]$. A two-dimensional rectangular area is also referred to as a strip. The width $w(n)$ can be adjusted to obtain different width to length ratios. More realistic network scenarios may be approximated by a combination of different strip shapes. We consider the static sensor network scenario where sensors do not move after the initial deployment. We assume that sensor nodes are randomly distributed according to a Poisson point process of density λ. Thus, the expected number of nodes in the network is $\lambda n w(n)$.

The widely used Boolean sensing model is adopted where each sensor has a certain sensing range, r. A sensor can only sense the environment and detect intruders within its sensing area. A location is said to be covered by a sensor if it lies within the sensor's sensing area. The space is thus partitioned into two regions, the covered region, which is the region covered by at least one sensor, and the vacant region, which is the complement of the covered region. In practice, sensing areas are never perfect disks. However, the disk model can provide lower and upper bounds for realistic irregular sensing areas [18].

Two sensors at locations X_i and X_j are connected if the sensing areas of the two sensors overlap, or equivalently, if $|X_i - X_j| \leq 2r$, where $|X_i - X_j|$ is the distance between the two sensors. A sensor barrier is defined to be a connected component of sensors that intersect both of the left and right boundaries of the strip. An intruder cannot go through a sensor barrier without being detected, since it will need to go through the sensing area of sensors and thus will be detected.

A *crossing path* is a path that connects one side of the region to the opposite side, where the ingress point and the egress point reside on two opposite sides of the region. For a two-dimensional strip, we assume that the intruders attempt to cross the width of the strip.

The strength of the barrier coverage of a sensor network can be measured by the number of times that an intruder is detected when traversing along a crossing path. A path is said to be k-covered if it intercepts at least k distinct sensors. An event is said to occur with high probability (w.h.p.) if its probability approaches 1 as $n \to \infty$.

Definition 1 *A sensor network is said to be k-barrier covered if*

$$P(\textit{any crossing path is } k\textit{-covered}) = 1 w.h.p. \tag{1}$$

3.2 Critical Conditions

The critical conditions for the existence and strength of barrier coverage is presented as follows:

Theorem 1 *Consider a sensor network deployed on a two-dimensional rectangular area* $A_{2\text{-}dim\ strip} = [0, n] \times [0, w(n)]$*, where sensors are distributed according to a Poisson point process with density* λ*.*

- *If* $w(n) = \Omega(\log n)$*, the network is barrier covered w.h.p. when the sensor density reaches a certain value. There exists a positive constant* β *such that w.h.p. there exist* $\beta w(n)$ *disjoint horizontal sensor barriers crossing the strip.*
- *If* $w(n) = o(\log n)$*, the network has no barrier coverage w.h.p. regardless what the sensor density is in the underlying sensor network. That is, w.h.p. there exist crossing paths that an intruder can cross the strip without being detected.*

It follows from Theorem 1 that the existence and strength of the barrier coverage depends on the width-to-length ratio of the strip region. The critical condition for barrier coverage to exist is when the width of the strip becomes asymptotically larger than the logarithm of the length, i.e., $w(n) = \Omega(\log n)$. The number of horizontally connected sensor clusters (barriers) is proportional to the width of the strip by a constant factor. On the other hand, when the width is asymptotically smaller than the logarithm of the width, there is no barrier coverage, i.e., there exist crossing paths that an intruder can cross the strip without being detected.

If $w(n) = \Omega(\log n)$, the network area can be divided into horizontal rectangles R_n of size $n \times \kappa r \log \frac{n}{r}$, for some $\kappa > 0$. It will become clear later from the proof of Theorem 1 that there exist $\beta\kappa \log \frac{n}{r}$ disjoint barriers in each rectangle and hence $\beta w(n)/r$ disjoint barriers in the whole strip, where $\beta = 1 - \frac{2(\kappa \log 6 + 2)}{\lambda r^2}$. This result can be used to answer a number of sensor deployment questions:

- How many barriers are present in the underlying sensor network?
- How does the number of barriers grow if more sensors can be added?
- What is the minimum number of sensors needed to achieve a given strong barrier coverage?

Proof of Theorem 1 We first convert the barrier coverage problem to a bond percolation model and use the results presented in [33] to complete the proof. Barrier coverage of a strip sensor network is directly related to the number of disjoint connected sensor clusters that cross the width of the strip horizontally. Two sensors are connected if their sensing areas overlap. Each of such sensor clusters acts as "trip wire" that can detect any crossing intruders. As in [33], we construct a bond percolation model to obtain the number of disjoint sensor clusters crossing the length of the strip.

We divide the area into squares of equal size, where the length of each side $d = r/\sqrt{2}$, as depicted in Fig. 2. By adjusting sensor density λ, we can adjust the probability that a square contains at least one sensor:

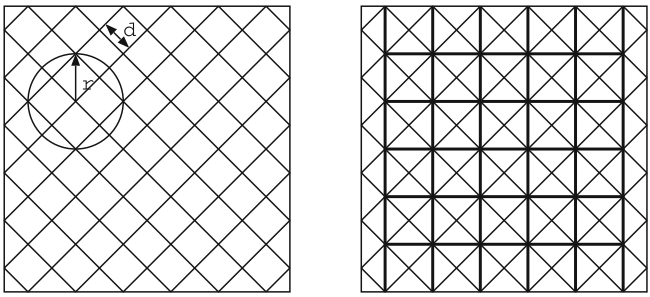


Fig. 2 Construction of the bond percolation model

$$\begin{aligned} p &= P(\text{a square contains at least one sensor}) \\ &= 1 - e^{-\lambda d^2} = 1 - e^{-c^2}, \end{aligned}$$

where $c^2 = \lambda d^2 = \lambda r^2/2$.

A square is said to be *open* if it is occupied by at least one sensor, and *closed* otherwise. Since the side length of each square is $r/\sqrt{2}$, the whole square will be covered by a sensor if it is *open*. Obviously, if two adjacent squares are both occupied by sensors, the sensing areas of the sensor would overlap and no intruder can cross between the two squares without being detected.

The above construction can now be mapped to a discrete bond percolation model as follows. Horizontal edges are added across half of the squares while vertical edges are added across others, as shown in the right-hand side of Fig. 2. This construction results in a grid of horizontal and vertical edges. A path consists of a sequence of consecutive edges. A path is said to be *open* or *closed* if it contains only open or closed edges respectively. Since the squares along an open path are all completely covered by sensor, a crossing open path from left to right of the strip acts as a barrier (or trip wire) that can detect any crossing intruders. The strength of the strong barrier coverage of a strip sensor network thus depends on the number of disjoint open paths.

If $w(n) = \Omega(\log n)$, we can divide the network area into horizontal rectangles R_n of size $n \times \kappa r \log \frac{n}{r}$, for some $\kappa > 0$. There are $\frac{w(n)}{\kappa r \log \frac{n}{r}}$ such rectangles. Let $m = \frac{n}{r}$, each rectangle R_n is of lattice size $m \times \kappa \log m$ in the bond percolation model, as illustrated in Fig. 3. The following lemma gives the number of disjoint open paths that cross each rectangle. The proof is similar to the proof of Theorem 3 in [33], which will be omitted here.

Lemma 1 *For any $\kappa > 0$, if $\lambda > 2(\log 6 + 2/\kappa)/r^2$, there exists a strictly positive constant $\beta(c, \kappa)$ such that w.h.p. there exist $\beta\kappa \log m = \beta\kappa \log \frac{n}{r}$ disjoint sensor clusters that cross each Rectangle R_n from left to right.*

For each rectangle R_n of width $\kappa r \log \frac{n}{r}$, there exist $\beta\kappa \log \frac{n}{r}$ disjoint sensor clusters that cross the rectangle. So the total number of such disjoint sensor clusters is $\beta w(n)/r$, which is linearly proportional to the width of the strip.

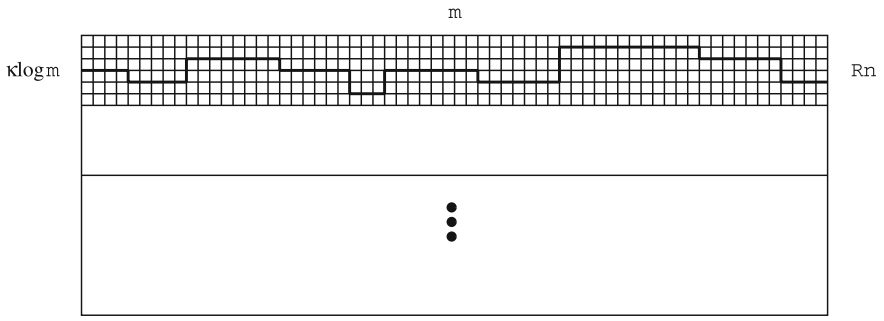


Fig. 3 The strip network is divided into $\frac{w(n)}{\kappa r \log \frac{n}{r}}$ horizontal rectangles of lattice size $m \times \kappa \log m$ where $m = \frac{n}{r}$. A *left* to *right* crossing of Rectangle R_n is shown

If $w(n) = o(\log n)$, a simple adaptation of Theorem 11.55 in [34], page 304, establishes that the probability that there is a path connecting the left and right sides of the strip is zero as $n \to \infty$, and therefore excludes the existence of the sensor clusters that cross the strip. There exist crossing paths that do not intercept any sensor such that intruders can cross the strip without being detected. □

The above barrier coverage results are based on the extended network model where node density is kept constant while the network size increases to infinity. In the previous study of critical condition for weak barrier coverage [25], the results are based on a network model of dimension $s \times (1/s)$ where the area of the network is kept constant while $s \to \infty$. To better link our results to those of [25], our network model can be rescaled to yield the critical conditions for barrier coverage of the $s \times 1/s$ network model. The result is presented in the following corollary.

Corollary 1 *Consider a two-dimensional strip sensor network of size $s \times 1/s$ where sensors are randomly distributed according to a Poisson point process of density λ and each sensor covers a disk of radius r. There exists $\theta > 0$, if $\lambda = \theta s^2 \log^2 s$ and the radius $r = 1/(s \log s)$, the strip is barrier covered as $s \to \infty$. The number of barriers is of order* $\log s$.

Proof Based on Lemma 1, denoting $m = n/r$, if $\lambda > 2(\log 6 + 2/\kappa)/r^2$ for some $\kappa > 0$, there exist a total number of $\beta\kappa \log m$ barriers in a rectangle region R_n of size $n \times \kappa r \log m$.

To establish the barrier coverage result in the network model R_s of size $s \times 1/s$, we let the length to width ratios of the rectangle R_n and R_s to be asymptotically the same, i.e., $n/\log n = s^2$. Therefore, the two network models have the same length to width radio by transformation $s = \sqrt{n/\log n}$, or conversely, $n = s^2 \log s$.

Now we need to rescale the sensor density and sensing radius in R_n to R_s. In R_n, the rectangle has length n and constant sensor density and radius. In R_s, the length is $s = \sqrt{n/\log n}$. Both the length and width of rectangle R_n are scaled by a factor of $s/n = s/(s^2 \log s) = 1/(s \log s)$ in R_s. Therefore, the sensor density should be

scaled by a factor of $s^2 \log^2 s$ and the sensing radius should be rescaled by a factor of $1/(s \log s)$. The number of barriers is on the order of $\log s$. □

3.3 Barrier Construction

Typical wireless sensors are powered by batteries and thus are energy stringent. It is therefore important to schedule sensors so that at any given moment there are just enough active sensors to cover the barrier. Other sensors will be set to the sleep mode to save energy for future use. This way, the operation lifetime of the network can be prolonged. Different from the sensor scheduling for area coverage, scheduling sensors for barrier coverage requires that sensors on the same barrier be synchronized to wake up or to sleep simultaneously. Otherwise the barrier will contain holes, which defeats the objective of the barrier coverage. Hence, it is important to find sets of sensors so that each of which forms a disjoint barrier. These sets of sensors can then take turn to form a barrier. Moreover, we want to find a scheduling to provide barrier coverage with low communication overhead and computation cost.

In [25], the authors show that whether a sensor network is strongly k-barrier covered cannot be determined using local algorithms. They convert the k-barrier coverage testing problem to the k-connectivity testing problem and refer to [35] for the best known global algorithms, which incur a time complexity of $O(k^2|V|)$ for a graph of $|V|$ nodes. To use the algorithm, each sensor node must broadcast its neighbor information to the whole network to construct the connectivity graph of the network. For a connected graph $G(V, E)$ of a sensor network, the message complexity (communication overhead) is $O(|V||E|)$ if the location of each node is broadcast to all of the other nodes, and the end-to-end delay of each message is proportional to the length of the strip. The communication overhead and delay can be formidably high for a large sensor network.

To reduce delay, communication overhead, and computation costs for finding disjoint barriers in a large sensor network, we cover the region to be protected by strips and divides each strip into small segments interleaved by thin vertical strips, as illustrated in Fig. 4. Each segment computes "horizontal" barriers and each vertical strip computes both "horizontal" and "vertical barrier" (of course these barriers do not have to be on straight lines). Horizontal barriers in strip segments are connected by vertical barriers in vertical strips to provide continuous barrier coverage across the whole network.

We first present an algorithm to find all disjoint barriers in a strip segment. Each node broadcasts its location and sensing range to all the sensor nodes in the segment. Alternately, each segment can select a node as the delegate for the entire segment and each node sends its location and sensing range to the delegate node. The location of a sensor node can be obtained by on-board GPS device or computed using a node localization scheme. After receiving the location information from other nodes in the segment, each node (or the delegate node) constructs a flow network $G(V \cup \{s, d\}, E)$ by making each sensor node a vertex in V with a vertex capacity of 1. For any two

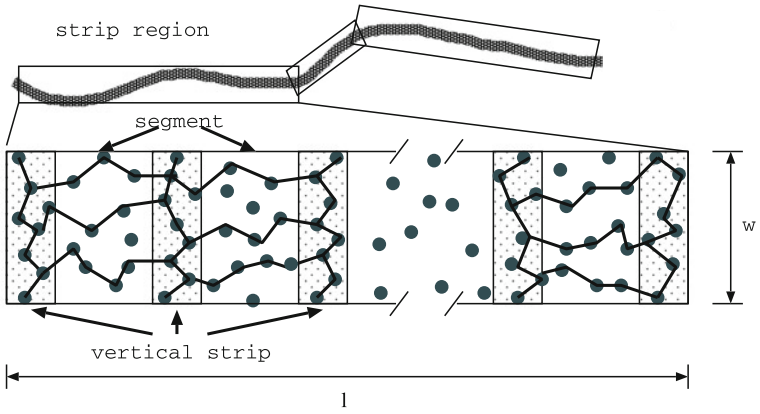


Fig. 4 The original strip is divided into small segments interleaved by thin vertical strips. Each vertical strip finds its horizontal and vertical barriers. Each segment finds the local horizontal barriers intersecting the vertical barriers on both ends. These local horizontal barriers are connected by vertical barriers so continuous barrier coverage across the whole strip is ensured. Each *dot* represents the location of a sensor. For conciseness, sensing areas of sensors are not shown in this figure

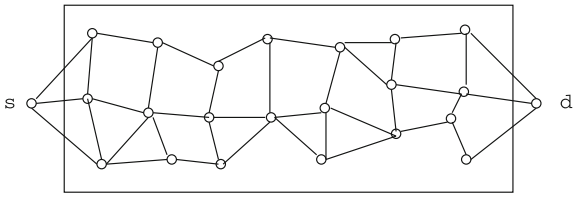


Fig. 5 Construction of flow networks. Sensors whose sensing areas intersect with the *left* and *right* boundaries are connected to s and d, respectively. Each edge and vertex is of capacity 1. The maximum flow from s to d gives the number of disjoint barriers. Sensors on the same flow path form a barrier

vertices u and v in V, if their sensing areas overlap, connect them with an edge capacity of 1. We add two nodes s and d. For each node $u \in V$, if its sensing area intersects with the left boundary of the segment, connect s to u with an edge capacity of 1; if its sensing area intersects with the right boundary of the segment, connect u to d with an edge capacity of 1. Figure 5 illustrates the construction of a flow network. The vertex capacity is used to ensure that each sensor node can only be used at most once in a barrier when finding a maximum flow from s to d. This flow network can be transformed to a traditional flow network by replacing each node $u \in V$ with two nodes u', u'' and between them a new edge (u', u'') of capacity 1, where u' has all the incoming edges of u, and u'' has all the outgoing edges of u. We then use a standard algorithm (e.g., Edmond-Karp or the relabel-to-front algorithms [36]) to find a maximum flow from s to d and all the paths used in the maximum flow. Based on the above construction, nodes of the same path form a barrier that connects the

left and right boundaries of the segment, and the maximum flow is the number of available disjoint sensor barriers.

The complexity of the relabel-to-front algorithm is $O(|V|^3)$. Since the number of sensor nodes deployed in a strip segment is much smaller than the number of sensor nodes deployed in the entire sensor field, the computational complexity would be much more manageable by sensor nodes. This algorithm can be easily modified to find all vertical barriers in a vertical strip by connecting s to all sensors whose sensing areas intersect with the top boundary and connecting d to all sensors whose sensing areas intersect with the bottom boundary. We refer to this algorithm as *ComputeBarriers*.

We now describe the divide-and-conquer approach to constructing disjoint barriers in a large strip sensor network.

Divide-and-Conquer Algorithm to Construct Barrier Coverage

1. Divide the given (curly) strip into small segments interleaved by thin vertical strips. The length of each vertical strip is $w(n)$, the width of the original strip. The width of each vertical strip is chosen to be of the order $\log w(n)$ such that there exist $\Theta(\log w(n))$ disjoint barriers crossing the vertical strip according to Theorem 1.
2. In each vertical strip, sensor nodes use *ComputeBarriers* to find all of the disjoint vertical barriers and the horizontal barriers that connect the vertical barriers together. This computation is carried out in each vertical strip independently.
3. For each strip segment, use *ComputeBarriers* to find disjoint horizontal barriers intersecting the vertical barriers on both ends of the segment. This computation is carried out in each strip segment independently in parallel.

In the above barrier construction process, each segment and vertical strip independently computes the horizontal barriers. These horizontal barriers are connected by vertical barriers in the neighboring vertical strips to provide global barrier coverage. This ensures that there is no gap between the horizontal barriers; so continuous barrier coverage across the whole strip is provided.

The *ComputeBarriers* algorithm finds all barriers in each strip segment and each vertical strip. If only k disjoint barriers are required, we can activate k horizontal barriers in each segment and in each vertical strip and rotate the active-duty barriers among all available barriers. Also, the vertical strips can be moved in a sliding window fashion to avoid the overuse of the same vertical barriers. The barrier rotation process and sliding vertical barrier scheme will balance the power consumption among sensors and hence extend the network lifetime.

Compared to the centralized approach that computes barriers for the whole strip, the above divide-and-conquer approach has the following advantages:

- *Lower communication overhead and computation costs.* By dividing the large network area into small segments, the message delay, communication overhead, and computation cost can be significantly reduced. The location and sensing area information of a sensor node only need to be broadcast within the strip segment (or within the thin vertical strip) where the node is located, resulting in a smaller

delay and communication overhead compared to the whole network broadcasting. For a connected graph G(V, E), the communication overhead (for location information broadcast) and computation complexity of the *ComputeBarriers* algorithm is $O(|V||E|)$, and $O(|V|^3)$, respectively. If the original strip is divided into n_s segments interleaved by thin vertical strips, each segment contains less than $|V|/n_s$ nodes and $O(|V|/n_s)$ links. The communication overhead is $O(|V|^2/n_s^2)$, a n_s^2 factor reduction from the centralized approach. The computation complexity is $O(|V|^3/n_s^3)$, a n_s^3 factor reduction from the centralized approach.

- *Improved robustness of the barrier coverage.* In a centralized approach which constructs global horizontal barriers for the whole strip, a horizontal sensor barrier could be broken if some nodes on the barrier fail, or become compromised or displaced by adversaries. In our divide-and-conquer approach, the original strip is divided into segments by interleaving vertical barriers. In case of node failure, these vertical barriers act as "firewalls" that prevent intruders from moving from its current segment to adjacent segments. This limits the barrier damages within the local segment and hence improving the robustness of the barrier coverage. A scenario of improved robustness with vertical barriers is illustrated in Fig. 6. The overhead of this approach is to compute the vertical barriers and use those nodes during operation.
- *Strengthened local barrier coverage.* By dividing the original strip into small segments and computing barriers in each segment, a larger number of local horizontal barriers will be found in each segment than for the whole strip. These local barriers are not necessarily part of the global barriers for the whole strip, whose number remains unchanged. Since adjacent segments are blocked by interleaving vertical barriers, a larger number of local barriers results in a strengthened local barrier coverage for each segment. Simulation results confirms that there is significant improvement of local barrier coverage in each segment over global barrier coverage.

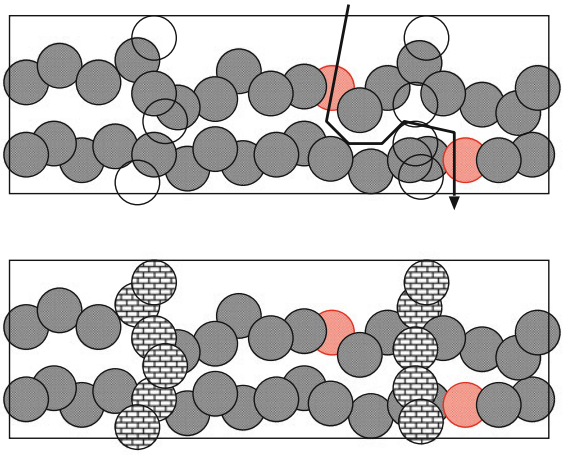


Fig. 6 Improved robustness for barrier coverage. Vertical barriers serve as "firewalls" that prevent intruders from moving from one segment to adjacent segments, resulting in improved robustness in case of barrier failures. *Shaded nodes* indicate activated sensors. *Red* (*lightly shaded*) *nodes* indicate failed sensors. *Non-shaded nodes* indicate non-activated sensors

3.4 Simulation Results

In the simulation, sensor nodes are distributed into the network of size $l \times w$ according to a two-dimensional Poisson point process of density λ. The mean number of nodes is $m = \lambda lw$. Each sensor has a sensing range of r. By varying the network parameters, λ, l, w, r, we can obtain a wide range of network scenarios. In each simulation, *ComputeBarriers* algorithm is used to find the number of disjoint barriers in the network. For every network scenario, the simulation is repeated 500 times to compute the mean values. The corresponding standard deviations are relatively small and not plotted.

In the experiments, we set the length of the region to be $l = 10{,}000\,\text{m}$ and the sensor's sensing range to be $r = 10\,\text{m}$. The width of the strip is varied from 50 to 1,000 m. This is repeated for different node densities $\lambda = 0.005, 0.0075$, and 0.01 nodes per unit area.

Figure 7 shows the number of disjoint horizontal barriers as a function of the strip width. It can be observed that for each of the node densities, there exists a critical width. The horizontal barriers only start to emerge when the strip width is larger

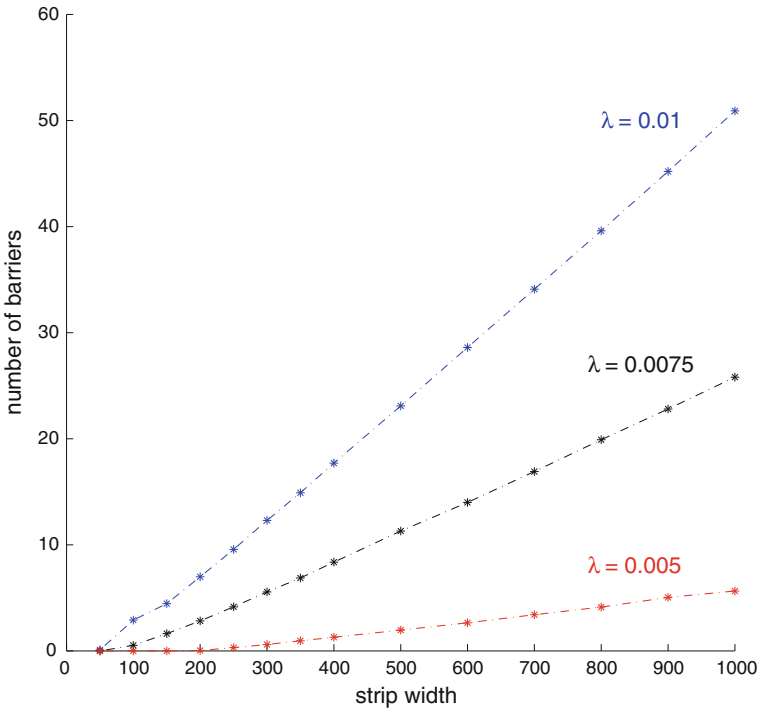


Fig. 7 Critical conditions for barrier coverage. Horizontal barriers start to appear only when the width is large enough. The number of barriers increase linearly with the width. For a given width, the network is barrier covered when the density is large enough

than the critical value. The larger the node density, the smaller the critical width beyond which horizontal barriers start to appear. As the width increases, the number of disjoint horizontal barriers at each node density increases linearly. Also, it can also be observed that for a given width, the network is barrier covered only when the density is large enough. These observations are consistent with the results in Theorem 1.

In the divide-and-conquer barrier construction algorithm, the whole strip region is divided into small segments interleaved by thin vertical strips. We first compute the vertical barriers in the vertical strips and then compute the horizontal barriers in each segment connected by the vertical barriers in the neighboring vertical strips. Compared to the original strip region, each segment has a larger width-to-length ratio with the same node density. Based on the results in Sect. 3.2, a larger number of local barriers will be found in each segment than the global barriers for the original strip. This will result in a strengthened barrier coverage for each segment.

In the simulation, we consider network scenarios of size 10,000 m $\times$ 250 m and node density $\lambda = 0.005, 0.0075$, and 0.01. The original strip is divided into ten segments interleaved by vertical strips. The length of each vertical strip is varied from 20 to 350 m. The length of each segment is set accordingly. We measure the improvement of barrier strength in each segment over the centralized approach by the *barrier improvement ratio*, defined to be the number of horizontal barriers in each segment divided by the number of global barriers for the whole strip.

Figure 8 shows how barrier improvement ratio changes with the width of the vertical strip. It can be observed that the number of barriers immediately increases as soon as the vertical strips are activated in our divide-and-conquer approach. The barrier improvement ratio continues to increase as the width of the thin vertical strip increases, and quickly levels off after some point. This is because as the width of the vertical strips increases, there will be more vertical barriers in each vertical strip, and thus a larger number of local barriers in each segment will be connected by these vertical barriers. However, after a certain point, most of the local barriers are already connected by vertical barriers. As a result, the barrier improvement ratio levels off. For node density $\lambda = 0.01$, the average number of global barriers for the whole strip in the centralized approach is 9.6. In our divide-and-conquer approach, with a vertical strip width of 100 m, the average number of horizontal barriers in each segment reaches 42.7, a more than four-fold increase over the centralized approach. The barrier improvement ratio is even more significant for smaller node densities, for example, the improvement ratio is close to 20 for node density $\lambda = 0.005$. This is because there are fewer global barriers for smaller node densities, allowing more room for improvement in each segment.

Compared to the centralized approach, the overhead of the divide-and-conquer approach is the employment of vertical barriers. Figure 9 shows the average number of vertical barriers for the above network scenarios: network of size 10,000 m $\times$ 250 m at node density $\lambda = 0.005$, 0.0075, and 0.01. The number of vertical barriers increases linearly with the width of each vertical strip. But we do not need to make the strip width too large and employ a large number of vertical barriers. Based on the observations for Fig. 8, the barrier improvement ratio quickly levels off after the width of

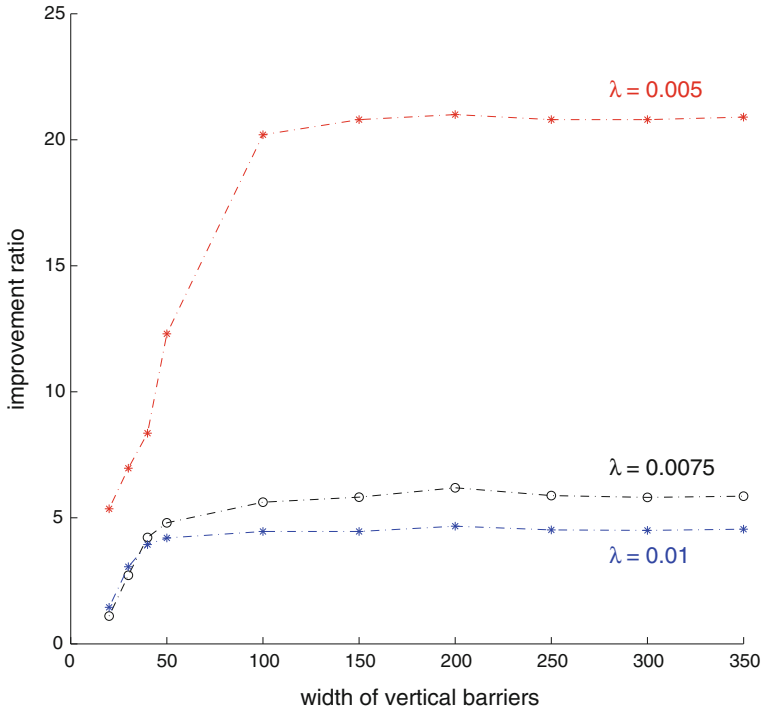


Fig. 8 Barrier improvement ratio. Each segment has more local barriers than the global barriers for the whole strip region. These local barriers in each segment are connected by neighboring vertical barriers to provide global barrier coverage

each vertical strip reaches a certain point. This provides a guideline to choose the width of the vertical strips. The proper width for a given network scenario can be obtained by simulation before the application of our algorithm. For example, the barrier improvement ratio saturates after the width of the vertical strips reaches 150 m, at which point the number of vertical barriers is less than 10 for all three densities.

4 Lined-Based Sensor Deployment

The choice of sensor deployment strategies has direct impact on barrier coverage. While manually placing sensors side by side along straight lines across the region yields the most efficient barrier coverage, it is infeasible to be implemented in many application scenarios such as hostile environments and hard-to-reach areas. A widely used random sensor deployment model assumes that sensors are distributed in a vast region uniformly at random and is approximated by a Poisson point process. For this model, it has been shown that certain conditions have to be met and the deployment can be inefficient for achieving barrier coverage [1]. Moreover, while the model

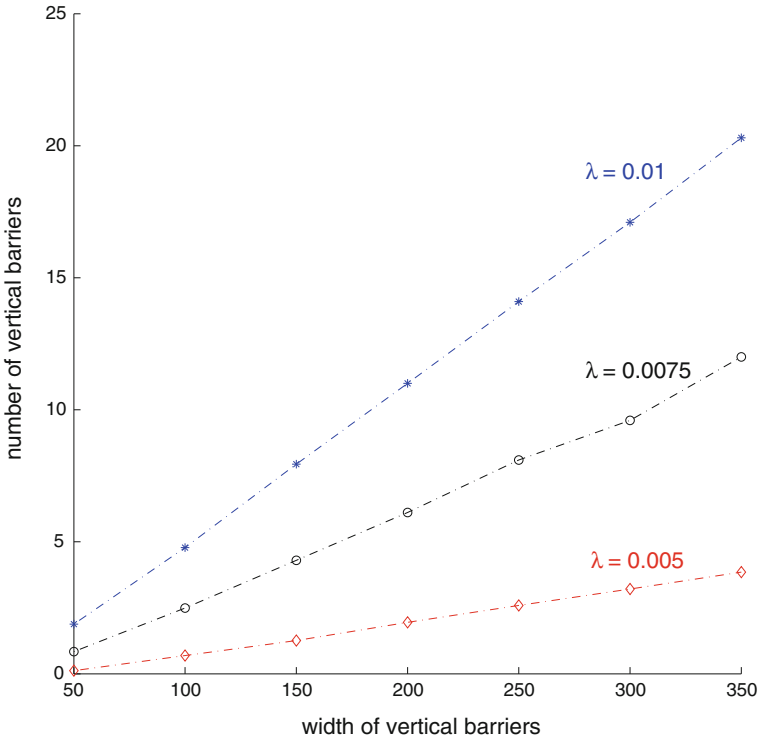


Fig. 9 Number of vertical barriers in each vertical strip

may be appropriate for some deployment strategies, the uniform distribution of sensors does not capture the sensor location distribution under many other deployment strategies.

When deploying sensors to monitor boundaries of battlefields or country borders with complex terrains, a viable approach is to drop a large number of sensors from aircraft along predetermined routes [2, 3]. When dispersed from an aircraft, sensors will be scattered and centered around the deployed line with some random offsets because of mechanical inaccuracy, wind, terrain characteristics, and other environmental factors. The line-based sensor deployment strategy is of particular interest to military and homeland security applications. We assume that the random offset of each sensor from its target landing point follows a normal distribution. For convenience, we refer to this type of distribution as *line-based normal random offset distribution*, or LNRO for short hereafter.

In this section we first present the assumptions and the corresponding analytical results for the probability of barrier coverage under LNRO. We then validate the analysis results via simulations, and compare the barrier coverage under LNRO with those under two-dimensional and one-dimensional Poisson point process models.

4.1 Line-Based Sensor Deployment Model

The sensors are deployed in a two-dimensional rectangular area of length l and width h (see Fig. 10). Each node is assumed to know its own location (x, y), which may be achieved by using an on-board GPS unit or other localization mechanisms.

We assume sensors are to be evenly deployed along the horizontal line $y = 0$. Let n be the number of sensors to be distributed along a given line. Let $\zeta = l/(n + 1)$. The horizontal coordinates of the ith target landing point is given by

$$x_i = \frac{il}{n+1} = i\zeta, \ 1 \le i \le n.$$

Because of mechanical inaccuracy, wind, terrain constraints, and other environmental factors, the actual landing point of each sensor will deviate from its target by a random offset. Denote by δ_i^x and δ_i^y the offset of sensor s_i in the horizontal and vertical directions, respectively. On the jth dropping line, the actual landing point of sensor s_i is thus $(x_i + \delta_i^x, \delta_i^y)$.

To simplify the analysis and provide insight, we assume that the random offsets are independently and identically distributed (i.i.d.) with a normal distribution of zero mean and variance σ^2, i.e.,

$$\delta_i^x, \delta_i^y \sim N(0, \sigma^2).$$

Note that the following analysis can be easily generalized to the case where δ_i^x and δ_i^y does not share the same standard deviation σ.

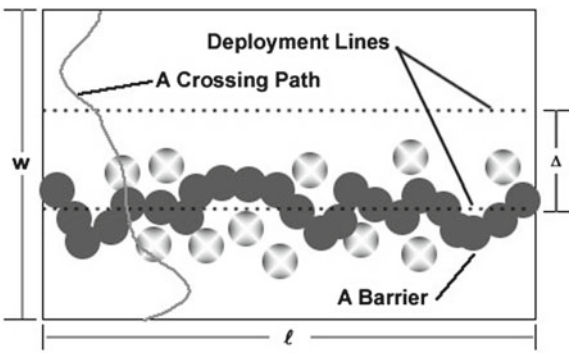


Fig. 10 Sensors are deployed along a straight line in a rectangular area of $l \times h$

4.2 Assumptions and Results

For the sake of obtaining analytical results to provide insights, the following assumptions are made:

1. Nodes are deployed so that the distance between adjacent targeted positions is within the sensing range of the two sensors. This means that $\rho > \kappa\zeta$, for some factor $\kappa > 1$ that will be specified later. Intuitively, we want to avoid the case where ρ is too close to $l/(n+1)$, for otherwise only small perturbation could create breaches in the barrier. We expect this assumption to be reasonable for many application scenarios as typical barrier coverage applications use sensors with large sensing ranges. For example, for MSP410CA wireless security system by XBow [37], the magnetic field sensors and infrared sensors have a sensing range of about 60 and 80 feet, respectively.
2. $\sigma \ll \zeta$. This assumption means that we still exert some control over the position of the nodes. The perturbation of the node position stays relatively small with respect to the gap between two nodes. We will see in the evaluation that our analysis stays valid with a standard deviation σ equal to 20 % of the targeted gap between two sensors ζ.

Based on the above assumptions, the main results are given as follows:

Theorem 2 *When assumption (1) and (2) are satisfied, the probability that barrier coverage exists for a single line covered by n sensors with coordinates* $(i\zeta + N(0,\sigma^2), N(0,\sigma^2))$, $1 \le i \le n$, *and sensing range* $r = \rho/2$ *is given by*

$$P(\text{BarrierCoverage}) \gtrsim \left[1 - Q_1\left(\frac{\zeta}{\sqrt{2}\sigma}, \frac{\rho}{\sqrt{2}\sigma}\right)\right]^{(n+1)}, \tag{2}$$

where

$$Q_1(s,t) = e^{-\frac{s^2+t^2}{2}} \sum_{k=0}^{\infty} \left(\frac{s}{t}\right)^k I_k(st) \tag{3}$$

and I_k *is the kth order modified Bessel function of the first kind.*

The proof of the theorem is provided in the appendix. In the next subsection it will be shown that, through a large number of numerical experiments, the lower bound given in inequality (2) is tight.

4.3 Probability of Barrier Coverage: Analysis Versus Simulation

Figure 11 plots the probability of barrier coverage of LNRO with standard deviation σ for the random offset. In the simulation, the length l is varied from 1,000 to 1,800 m. Nine nodes of sensing range $r = 100$ m are deployed along the line. Thus the

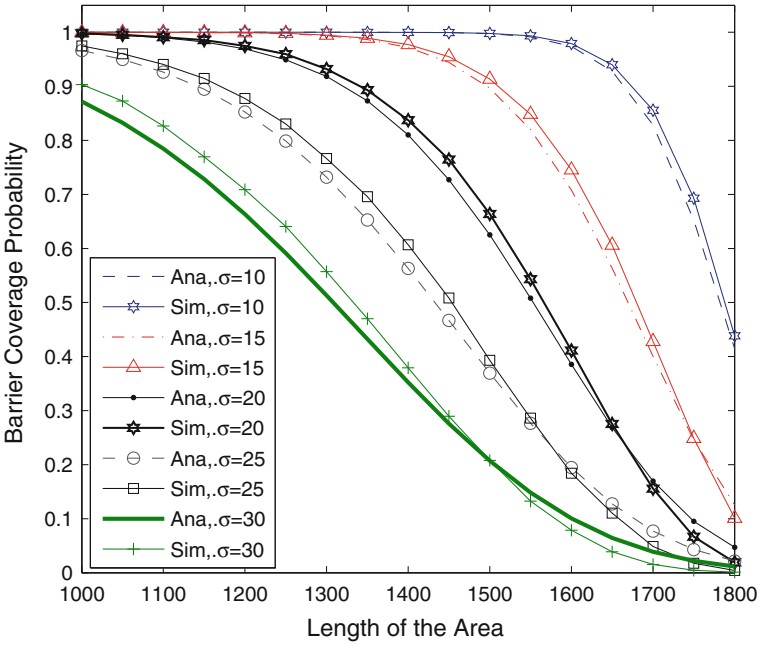


Fig. 11 Probability of barrier coverage for various area lengths with 9 nodes and $\rho = 200$. From *left* to *right*, $\sigma = 30, 25, 20, 15, 10$

connectivity radius is $\rho = 200$ m, and the distance between adjacent target positions, ζ, varies between 100 and 180 m, satisfying Assumption (1). We vary the standard deviation σ between 10 and 30, which meets the requirement of Assumption (2). The curves from left to right correspond to the reversed order of the chosen standard deviations, i.e., $\sigma = 30, 25, 20, 15$, and 10.

It can be observed that there is a good match between the simulation and analytical results. The match improves as the variance decreases. One can also verify that the analysis is indeed a lower bound as long as Assumption (1) is satisfied, i.e., as long as ρ stays larger than $\kappa\zeta$, where κ is in the range of 1.05 for $\sigma = 10$–1.33 for $\sigma = 30$.

The good match between the analysis and the simulation is insensitive to the number of nodes, as can be observed from Fig. 12, where the number of nodes has been increased to 99, and ρ reduced to 40 m. The corresponding ζ varies between 11 and 30. Recall that Assumption (2) requires that the standard deviation be a relatively small fraction of ζ. For the leftmost pair of curves in Fig. 12, ζ varies between 11 and 22, and the standard deviation is $\sigma = 8$. This breaks Assumption (2), and indeed, the match is poor between the analysis and the simulation. However, when we decrease σ to 5 and 3, simulation and analysis starts to track each other very well again.

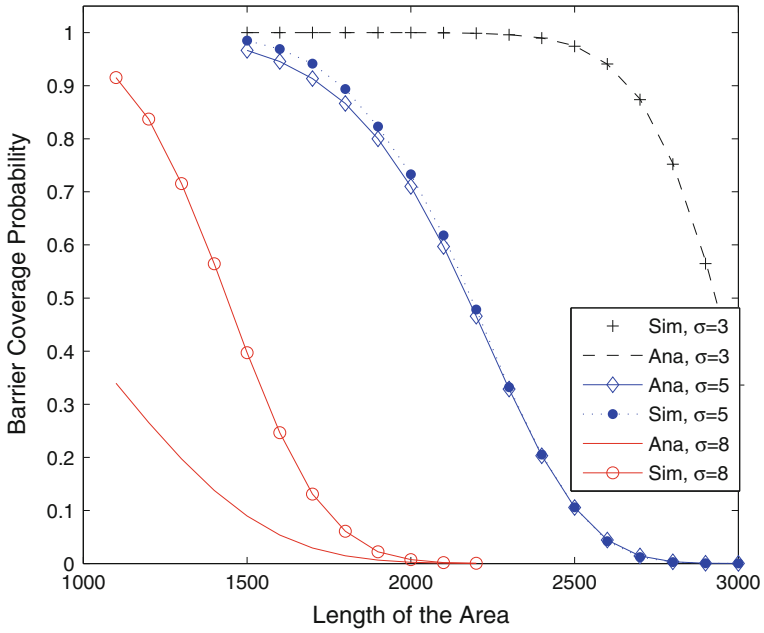


Fig. 12 Probability of barrier coverage for various area lengths with 99 nodes and $\rho = 40$. From *left* to *right*, $\sigma = 8, 5, 3$

4.4 Comparison with Two-Dimensional Poisson Model

Most previous studies on barrier coverage consider the scenario where nodes are distributed according to a Poisson process or a uniform distribution in the area to be covered. We now compare our results with the barrier coverage probability under a Poisson process.

In the case of uniform distribution, each sensor has the equal likelihood to be located at any point in the rectangle. Thus, the sensors are spread out rather evenly in the area. It has been proved in [1] that in the asymptotic case, barriers exist if and only if the width of the rectangle is larger than the logarithm of the length and at the same time the sensor density λ is greater than some critical value.

In the line-based deployment strategy with normally distributed random offsets, sensors are concentrated along the deployment line. The node density in the vertical direction forms a "bell" curve whose shape is determined by the variance of the normal distribution. Figure 13 illustrates the deployment layouts of the uniform distribution and line-based normal distribution, and the corresponding sensor densities in the vertical direction.

Compared to the uniform distribution, sensors are concentrated along deployment lines in the line-based deployment strategy, providing a better chance for barriers to be formed. To compare the barrier coverage of LNRO with that of the Poisson point

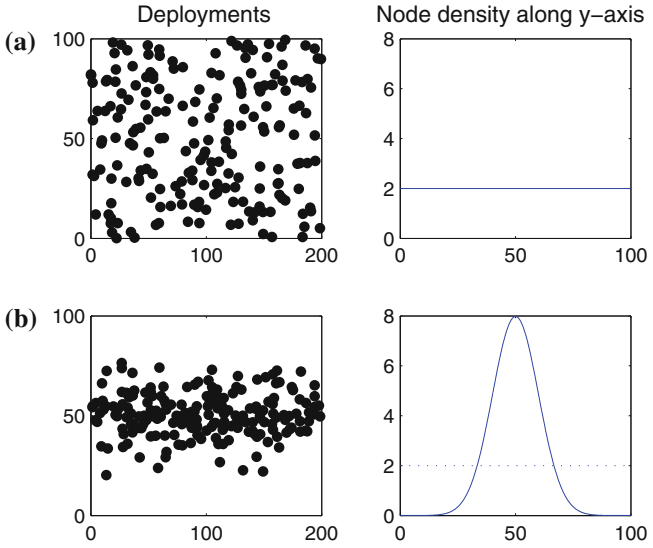


Fig. 13 Two hundred sensors are deployed under uniform random distribution and LNRO. The corresponding sensor densities along y-axis are plotted alongside with the sensor deployment scenarios. **a** Uniform distribution, $\lambda = 0.02$; **b** LNRO, $\sigma = 10$

process, we note that 99.7 % of the sensors fall within the distance of 3σ from the deployment line in LNRO, and so we choose the width of the rectangle for the uniform distribution to be 6σ for comparison.

For the Poisson point process deployment, the probability that the nodes provides barrier coverage is given by [38]. The main results are described as follows. Define a strip of width h and length l. Nodes are distributed according to a Poisson process with density λ and have a connectivity radius ρ. To make the comparison between the LNRO distribution and the Poisson point process fair, we set h to be equal to a multiple of σ.

Define the break density to be $I_{h,\rho,\lambda}$. With the proper definition, breaks in the coverage can be shown to follow a Poisson distribution as well, and thus the density of this Poisson process is $I_{h,\rho,\lambda}$. The probability that the strip provides barrier coverage is thus equal to the probability that there is no break, which is equal to:

$$P(\text{Barrier Coverage}) = e^{-lI_{h,\rho,\lambda}} \tag{4}$$

$I_{h,\rho,\lambda}$ can be approximated by:

$$\begin{aligned} I_{h,\rho,\lambda} &= \sqrt{\lambda} I_{h\sqrt{\lambda},\rho\sqrt{\lambda}} \quad \text{where} \\ I_{a,b} &= e^{-\alpha_a b - \beta_b} \text{ and} \\ \alpha_a &= a - 1.12794a^{-\frac{1}{3}} - 0.20a^{-\frac{5}{3}} \end{aligned}$$

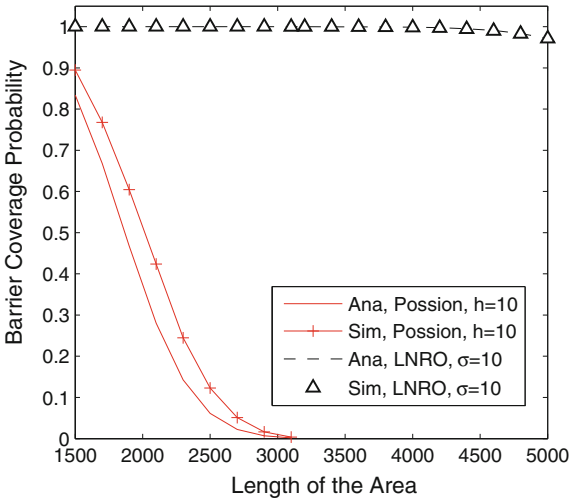


Fig. 14 Probability of barrier coverage for various area lengths with 99 nodes, $\rho = 100, h = 10$ for the Poisson deployment and $\sigma = 10$ for the LNRO deployment

$$\beta_b = -\frac{1}{3}\log(b) + 1.05116 + 0.27b^{-\frac{4}{3}} \tag{5}$$

One requirement to study the barrier coverage with a Poisson process is to assume that $\lambda > \lambda_c$, where λ_c is the critical density for percolation. In other words, λ should be in the supercritical regime. For a network of size $l \times h$, we have $\lambda = \frac{n}{hl}$. If we set $h = 6\sigma$, with $\sigma = 10$, then $\lambda = 0.00015$. The critical density, when $\rho = 200$, is $\lambda_c = 0.0772$. (The critical density depends on the connectivity radius in such manners that the average number of neighbors is equal to 4.5118 for the Gilbert model [39]). One would need to narrow down the strip to a width $h = \sigma/86$ to achieve super-criticality of the Poisson process. And indeed, either using Eq. (5) or simulating the deployment according to a Poisson process with $\lambda = 0.00015$, $h = 6\sigma$, $\sigma = 10$, $n = 9$ and $l = 1000$ shows that barrier coverage is achieved with a negligible probability.

Even in the supercritical regime, the probability of barrier coverage is very small for the Poisson process compared to that of the LNRO deployment. Figure 14 compares the LNRO and Poisson process in an area of length varying between 1,500 and 5,000 m. The LNRO deployment, with 99 nodes, $\rho = 100$ and a standard deviation of $\sigma = 10$ ensures barrier coverage with probability close to one, both in the analytical model and the simulation, across the whole range of l. The Poisson process, in a strip of size $l \times \sigma$ (note that the LNRO deployment is 97 % contained within a strip of size $l \times 6\sigma$, thus in a much wider strip), has on average 99 nodes with $\rho = 100$. The parameters are chosen so that λ is firmly supercritical (in the Gilbert model) for all values of l. Figure 14 shows both simulation and analysis from Eq. (5) for the Poisson deployment, with a much lower probability of barrier coverage, despite the favorable parameters.

4.5 Comparison with One-Dimensional Poisson Model

In LNRO, sensors are concentrated along the deployment line with random offsets. Thus, it is also interesting to compare the barrier coverage of LNRO with that of a strict line deployment where all sensors fall on the same line ($\sigma^y = 0$).

Figure 15 compares the probability of barrier coverage of LNRO with that of a line distribution according to a uniform line distribution. For a given line segment of length l, the corresponding probability density function (pdf) of a node location in the strict line deployment is $P(x) = 1/l$, when $x \in [0, l]$, and 0 otherwise. In the simulation, we set $\rho = 100$, $\sigma = 20$, $n = 50$, and l from 1000 to 5000. It can be observed that the barrier coverage of LNRO consistently outperforms that of single line uniform case. Reducing the variance in the y-dimension (σ_y) in LNRO will further increase the barrier coverage probability of LNRO, resulting in better performance over the single line uniform distribution. Simulations show a relatively similar probability of barrier coverage for a Poisson point process and for a uniform distribution with the same average number of nodes.

For the uniform distribution along a single line, a similar analysis to that of Theorem 2 shows that, under the Assumptions (1) and (2), the barrier coverage probability for the Poisson point process case is given by

$$P(\text{Barrier Coverage}) = \left(1 - e^{-\frac{n\rho}{l}}\right)^n. \tag{6}$$

We also consider a strict line deployment with Normal perturbation along the line. In this case we have $\sigma_y = 0$, and $x_i = i\zeta + N(0, \sigma_x)$. Similar analysis as for Theorem 2 shows that

$$P(\text{Barrier Coverage}) = \left(P(N(n/l, \sqrt{2}\sigma) < \rho)\right)^n$$

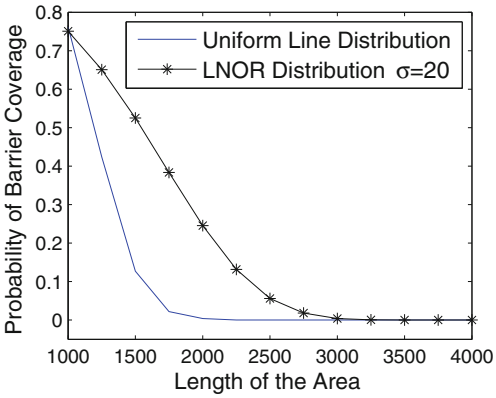


Fig. 15 Probability of barrier coverage for various area lengths with 50 nodes, $\rho = 100$, for a uniform sensor distribution on a line, and for a two-dimensional LNRO deployment with $\sigma = 20$

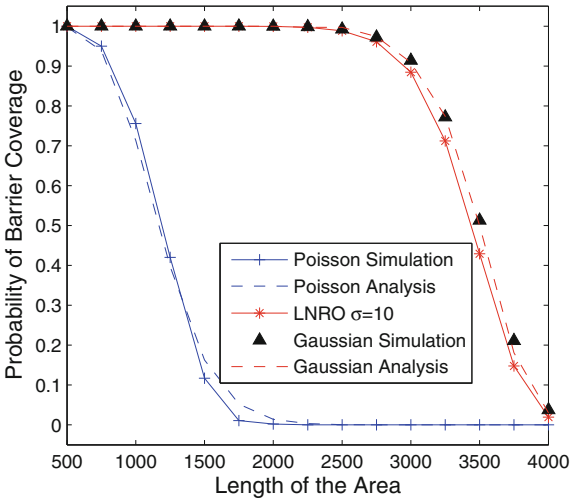


Fig. 16 Probability of barrier coverage for various area lengths with 50 nodes, $\rho = 100$, for a distribution of sensors on a line with Poisson distribution, for a regular distribution with a Normal (0,10) perturbation along the x-axis, and for a two-dimensional LNRO deployment with $\sigma = 10$

$$= \left(\frac{1}{2} + \frac{1}{2}\text{erf}\left(\frac{\rho - \zeta}{2\sigma}\right)\right)^n \tag{7}$$

Figure 16 plots the numerical results of Eqs. (6) and (7) with the corresponding simulation results. For reference, the LNRO case is also included. It can be observed that for both cases the analysis matches the simulation results very well. Also, the barrier coverage of LNRO and the line deployment with Normal perturbation are close to each other, both outperforming the line deployment with Poisson distribution.

5 Mobility Improves Barrier Coverage

Recently, there has been increasing interest in deploying mobile sensor networks, which can be extremely valuable in hostile environments such as battlefields and hazardous areas. Numerous mobile sensor platforms have been developed, including Packbot [40], Robomote [5], and Khepera [6], etc. As technologies advance, these mobile sensor platforms will become increasingly available and may be deployed on a large scale in practical applications in the future.

Most of the previous studies on the barrier coverage of wireless sensor networks consider constructing barriers with stationary sensors [1, 4, 21, 25, 26, 31]. While there has been much effort investigating how to move sensors to improve area coverage of a wireless sensor network [41–45], the impact of sensor mobility on barrier coverage has not been adequately explored. In this section, we describe the recent results on exploiting sensor mobility to improve the barrier coverage.

In [1], it has been shown that barrier coverage is difficult to achieve when sensors are randomly deployed. This is because a large fraction of sensors will not contribute

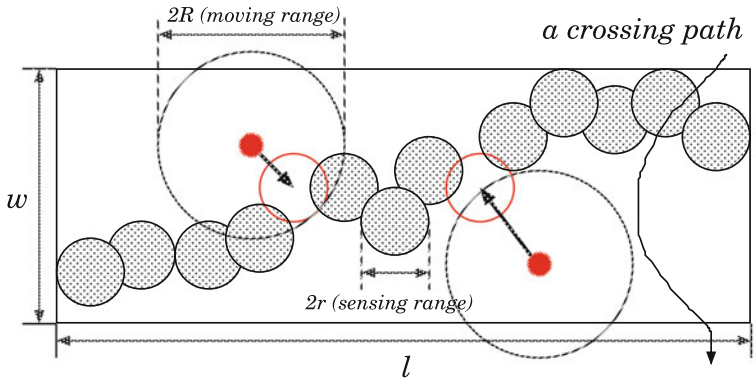


Fig. 17 Mobile sensors relocate themselves to improve the barrier coverage of the network. Sensors are deployed in rectangular region of $\ell \times w$. Sensors' sensing range is r, and mobile sensors' maximum moving range is R

to barrier coverage. In a mobile sensor network, after the initial deployment, mobile sensors can move to desired locations and connect with other un-utilized sensors to form new barriers, as illustrated in Fig. 17. Therefore, it is important to effectively exploit sensor mobility so as to improve barrier coverage. Otherwise, a sensor network deployment may not achieve its coverage goal and the sensors will be wasted.

While the potential improvement is promising, it is challenging to compute the desired location for each mobile sensor to move to, as well as to explore the performance potentials of sensor mobility, due to the non-local nature of the barrier coverage. For example, in Fig. 17, a barrier will be formed only when both mobile nodes are relocated to the desired locations as indicated. Moreover, another challenge of constructing barriers with mobile sensors is that existing mobile sensor platforms are often powered by small batteries which significantly limit the range of their movement. For instance, the on-board batteries of Robomote nodes only last for about 20 min in full motion. Given a typical speed of 15 cm/s, the range of movement is only about 180 m [5].

Given a network scenario (initial sensor deployment, number of mobile sensors, and their moving ranges), it is desirable to form the maximum number of disjoint barriers so as to provides effective and robust defense against intruders. The final barrier coverage is dictated by the sensor mobility scheme that determines the location that each mobile sensor should move to. A desirable sensor mobility scheme should take advantage of the existing network topology and efficiently improve barrier coverage under the limited mobility constraint.

Moving sensors to join a large local cluster may not create new barriers, as the cluster may not continue to cross the whole region. The creation of a new barrier often require multiple sensors to be relocated to certain locations respectively. However, a sensor cannot move to an arbitrary location due to its limited moving range. Moreover, while a mobile sensor may move to any location within its moving range, it can only contribute to at most one disjoint barrier.

The following problems are of particular interest: Where should each mobile sensor move to maximize the number of barriers that can be formed? How does the improvement depend on the number of available mobile sensors and their moving range? What are the fundamental limits of the barrier coverage using mobile sensors and what is the corresponding mobility requirement? Answers to these questions provide important insights into the design and performance of wireless sensor networks for barrier coverage.

5.1 Network Model

Two classes of sensor location distribution models have been considered in the previous studies: the uniform distribution model where sensors are deployed in a region uniformly at random, and the line-based model where sensors are deployed along certain lines [4]. Both models are valid and applicable for different application scenarios. For example, when sensors are dropped by an aircraft along its flying route, the sensor distribution follows the line-based model. If sensors are launched from artillery ordinance to an area uniformly at random, it may be approximated by the uniform distribution model. In this section we consider the impact of sensor mobility on the barrier coverage under both sensor location distribution models.

In the initial deployment, a combination of n stationary and m mobile sensors are distributed uniformly at random in a large two-dimensional rectangle area of size $\ell \times w$. Due to the application requirement of barrier coverage, the deployment region is often a thin strip area, for example, boundaries of a battlefield and perimeters of a critical infrastructure. The length of the rectangle is usually significantly larger than its width, i.e., $\ell \gg w$. In the asymptotic case, $\ell(n), w(n) \to \infty$ as $n, m \to \infty$, the initial sensor locations follow a Poisson point process. The densities of the stationary and mobile sensors are $\frac{n}{\ell(n)w(n)}$ and $\frac{m}{\ell(n)w(n)}$, respectively.

After the initial deployment, mobile sensors can relocate themselves. Due to power constraint, we assume each mobile sensor has a maximum moving range of R. As shown in Fig. 17, a mobile sensor, initially located at (x_0, y_0) can move to any location within a circle of radius R, i.e., any point (x, y) with $(x - x_0)^2 + (y - y_0)^2 \leq R^2$.

A challenge to using sensor mobility to improve barrier coverage is the *non-locality* nature of the problem. As shown in Fig. 18, based on the location information of nearby sensors, the mobile sensor (solid dots) on the right-hand side can choose to fill one of the two "gaps." However, without the global network topology information and coordinating with the other mobile sensor, it cannot make an informed decision to form a global barrier.

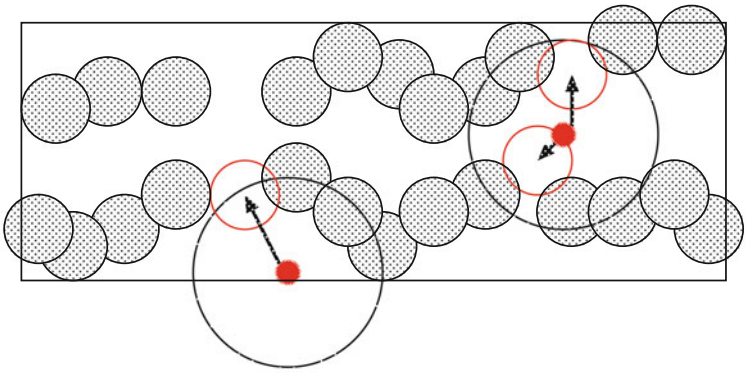


Fig. 18 Example for the non-locality of mobile barrier coverage problem

5.2 Minimum Required Moving Range

The section first investigates the fundamental limit of the barrier coverage that a mobile sensor network can provide, as well as the requirement on the sensor mobility to reach the limit. An efficient sensor movement scheme is then devised to provide maximum barrier coverage while minimizing the maximum moving distance among all sensors. Last, simulations are used to evaluate the performance of the sensor movement scheme.

Sensor movements are often powered by batteries (or local fuel reserve on robots), and normally individual sensors do not share power. Minimizing the total distance traveled by all sensors will minimize total energy cost but will not necessarily lead to balanced power consumption among sensors. On the other hand, minimizing the maximum of distance traveled by any sensor will balance the power consumption among sensors, thus prolonging the network lifetime. For network planning, it is important to find out the minimum required moving range of mobile sensors, as it decides the battery capacity needed to achieve barrier coverage in a mobile sensor network.

5.2.1 Analytical Results

Theorem 3 *When m mobile sensors are deployed in a rectangle area of size $\ell \times w$, the maximum number of horizontal barriers that can be formed is*

$$n_b = \lfloor \frac{2mr}{\ell} \rfloor \tag{8}$$

To achieve this limit, the expected minimum of the maximum moving distance among all mobile sensors is

$$d_m = \Theta(\sqrt{\ell r} + w) \ w.h.p. \tag{9}$$

Proof Each sensor covers a disk area of radius r. In a rectangle area of size $\ell \times w$, a horizontal barrier requires at least $\ell/2r$ sensors to be placed along a line side by side. Therefore, the maximum number of barriers that can be formed is

$$n_b = \lfloor \frac{m}{\ell/2r} \rfloor = \lfloor \frac{2mr}{\ell} \rfloor$$

Consider the following two-phase sensor movement scheme to form k ($k \leq n_b$) horizontal barriers. These k barriers may be evenly spaced out within the rectangle region, or be located at arbitrary vertical locations according to application requirements.

1. *First Phase* (*vertical movement*): sensors move vertically to evenly populate $k (k \leq n_b)$ horizontal lines. After movement, each line will have m/k sensors. The maximum moving distance along the vertical direction is $\Theta(w)$.
2. *Second Phase* (*horizontal movement*): sensors move horizontally to their assigned grid points positions (defined below as Y) along the lines.

In the second phase, sensors on each line are initially distributed uniformly at random. To form a barrier, they will need to be relocated to grid points of coordinates $(2i + 1)r, 0 \leq i \leq \ell/2r - 1$. Once every grid point is occupied by a sensor, there is no gap on the line and a barrier is created.

The sensor relocation can be considered as a *minimax grid matching problem* [46], where mobile sensors need to be perfectly matched to the grid points with the coordinates specified above.

Denote the initial sensor locations by

$$X = \{x_1, \ldots, x_{\ell/2r}\},$$

and the grid points on the line by

$$Y = \{y_i = (2i + 1)r\}_{i=0}^{\ell/2r-1}.$$

Let $L(X, Y)$ denote the minimum length such that there exists a perfect matching of the points in X to the grid points in Y for which the distance between every pair of matched points is at most $L(X, Y)$. In other words, $L(X, Y)$ is the minimum over all perfect matchings of the maximum distance between any pair of matched points, minimax matching length, and is thus called the *minimax matching length*.

From [46], for 1-dimensional case where n points are matched to the grid points within [0,1], the expected value of the minimax matching length is $\Theta(1/\sqrt{n})$, i.e., there are positive constants c and C such that

$$c \leq n^{1/2} E[L(X, Y)] \leq C.$$

Applying proper scaling in our model where a number of $\ell/2r$ points are matched to grid points within segment $[0, \ell]$, we have

$$E[L(X, Y)] = \Theta(\sqrt{\ell r}).$$

Combining the two moving phases, the expected minimax moving distance among all sensors is

$$d_m = \Theta(\sqrt{\ell r} + w) = \begin{cases} \sqrt{lr} & \text{if } w = O(\sqrt{lr}) \\ w & \text{if } w = \omega(\sqrt{lr}) \end{cases}$$

□

Discussions of results:

- In practice, sensors do not need to move in this two-phase (vertical then horizontal) fashion. They can directly move to the final locations in a straight line with a shortened distance. Nevertheless, the two-phase scheme can be used to compute the final location for each sensor. Based on the results, each sensor then moves to its final location in a straight line of distance

$$d = \Theta\left(\sqrt{lr + w^2}\right) = \begin{cases} \sqrt{lr} & \text{if } w = O(\sqrt{lr}) \\ w & \text{if } w = \omega(\sqrt{lr}) \end{cases}$$

  The asymptotic behavior of the minimax moving distance remains the same as the two-phase sensor mobility scheme. Therefore, the two-phase sensor mobility scheme is order optimal in achieving the maximum barrier coverage while minimizing the maximum moving distance among sensors.
- Depending on the relative magnitude of w and $\sqrt{lr}$, the minimax moving distance among all sensors is dominated by movement in different directions. Specifically, when $w = O(\sqrt{lr})$, the horizontal movement dominates the total moving distance. Otherwise, when $w = \omega(\sqrt{lr})$, the total moving distance is dominated by the vertical movement.
- If the goal is to minimize the total moving distance of all sensors, a similar matching problem, the *transportation problem*, can be considered. Denote $T(X, Y)$ as the minimum sum of the distance between matched pairs of points in X and Y. It is easily shown that $cn^{1/2} \leq E[T(X, Y)] \leq Cn^{1/2}$, giving the same asymptotic results as in the minimax grid matching problem [46].

5.2.2 Sensor Movement Scheme

We now present a sensor movement scheme that matches each mobile node to a grid point and minimizes the maximum moving distance among all sensors.

As shown in Fig. 19, each mobile sensor first move vertically to its projection on a pre-defined line in the first phase. In the second phase, given the set of mobile

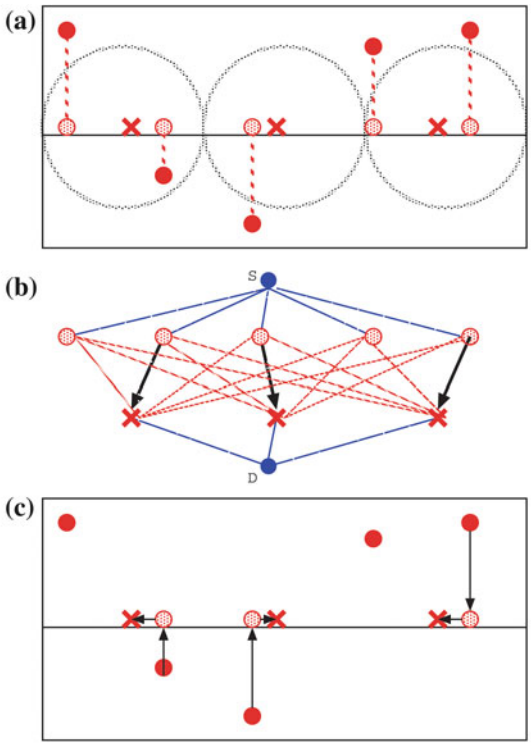


Fig. 19 The two-phase sensor movement scheme. Sensors first move vertically to a pre-specified line of defense. They are then assigned to the different grid points, move horizontally to their final locations, and form a barrier

sensors S, the grid points Y, and sensor's moving range d, the following algorithm computes if every grid point can be occupied by a mobile sensor under the sensor mobility constraint.

MAX- FLOW(X, Y, d)

1. Construct a bipartite graph $G(V, E)$ $(V = X \cup Y)$ as follows. Each vertex in X represents a mobile sensor, and each vertex in Y represents a grid point along the line. $E = \{(u, v), (v, u) | u \in X, v \in Y$, and $dist(u, v) < d\}$.
2. From $G(V, E)$, construct a flow graph $G^*(V^*, E^*)$ and assign capacity to each edge as follows: $\forall$ u $\in$ V, add u to V^*; $\forall$ (u, v) $\in E$, add (u, v) to E^*. Set $capacity(u, v) = 1$ if $u \in X$ and $v \in Y$, otherwise, set $capacity(u, v) = 0$. Add a virtual source node S to V^* , and $\forall$ u $\in S$, add an edge (S, u) to E^*, set $capacity(S, u) = 1$; add a virtual sink node D to V^*, and $\forall$ u $\in Y$ of $G(V, E)$, add an edge (u, D) to E^*, set capacity(u, D) = 1.
3. Use a maximum flow algorithm (e.g., Ford-Fulkerson [36]) to compute and return the maximum flow from S to D in G^*.

The Max-Flow algorithm above terminates in $O(VE^2)$ time. When the algorithm terminates, if the returned maximum flow from S to D equals the number of grid points, each grid point will be assigned a sensor and a barrier can be formed.

Otherwise, if the returned maximum flow is smaller than the number of grid points, there are not enough sensors to occupy all the grid points, i.e., some grid points will not be occupied by sensors.

Although all sensors are assumed to be mobile in this case to explore the limit of the barrier coverage that a mobile sensor network can provide, the Max-Flow algorithm can handle the scenario where there are both stationary and mobile sensors. A stationary sensor is equivalent to a mobile node with a zero moving range.

A binary search is then used to find the minimax moving distance among all sensors.

MINIMAX- MOVING- DISTANCE (X, Y)

```
1  last_success ← ℓ; last_fail ← 0; d ← ℓ
2  while (last_success - last_fail ≥ ϵ)
3        do f ← MAX- FLOW(X, Y, d)
4           if (f = SIZEOF(Y))
5              then last_success ← d
6              else last_fail ← d
7           d ← (last_fail + last_success)/2
8  return d
```

The above binary search-based Minimax-Moving-Distance algorithm terminates in $\Theta(\log \ell)$ iterations. When it terminates, it will return the minimum moving distance that allows every grid point to be occupied by a sensor. The ϵ in line 2 of MINIMAX-MOVING- DISTANCE is a termination threshold, representing the precision of the d obtained from this algorithm. In each iteration, MAX-FLOW is executed once, so the total running time of the sensor mobility scheme is $O(\log \ell\, V E^2)$.

To validate the analytical result, consider the scenario where m mobile nodes are randomly deployed in a rectangular area of length $2mr$ and width w. As discussed earlier, the maximum moving distance of the vertical movement is $\Theta(w)$. Here we focus on the minimax moving distance of the horizontal movement.

To demonstrate the performance of the above sensor mobility scheme, a greedy approach is use as a reference point. The greedy approach tries to assign the closest available mobile sensor to each grid point. Each time we randomly select a grid point that has not assigned a sensor, and assign the closest available mobile sensor to the grid point. This process is repeated until all the grid points are occupied.

Figure 20 compares the minimax moving distance of our scheme with that of the greedy algorithm. As the length of the field increases, the minimax moving distance of the greedy algorithm grows linearly, while in our proposed scheme the growth is sub-linear, resulting in a widening gap between the two approaches.

According to Theorem 3, the minimax moving distance (d_m) in our scheme is proportional to the square root of the length (ℓ), i.e., $d_m = \Theta(\ell^{0.5})$. This is confirmed by the regression results, as shown in Fig. 20. For example, the simulation results for the case $r = 20$ can be well fitted by $d_m = a\ell^b + c$, where $a = 5.8934$, $b = 0.4919$, and $c = -27.4322$. The 95 % confidence interval of b is [0.4749, 0.5090] (Fig. 21).

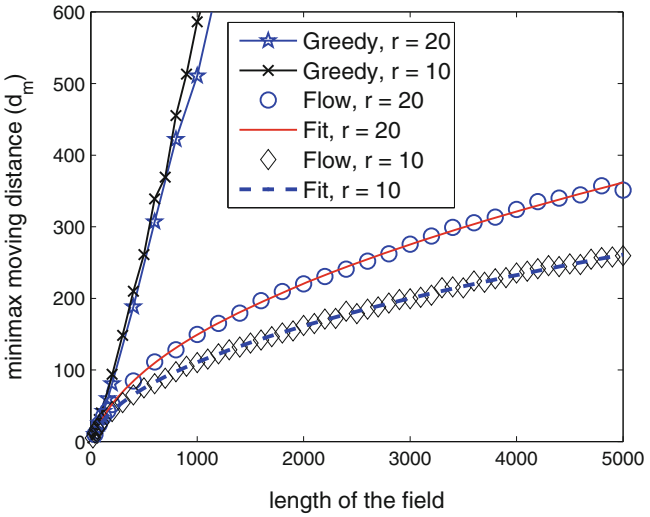


Fig. 20 Minimax moving distance to achieve barrier coverage

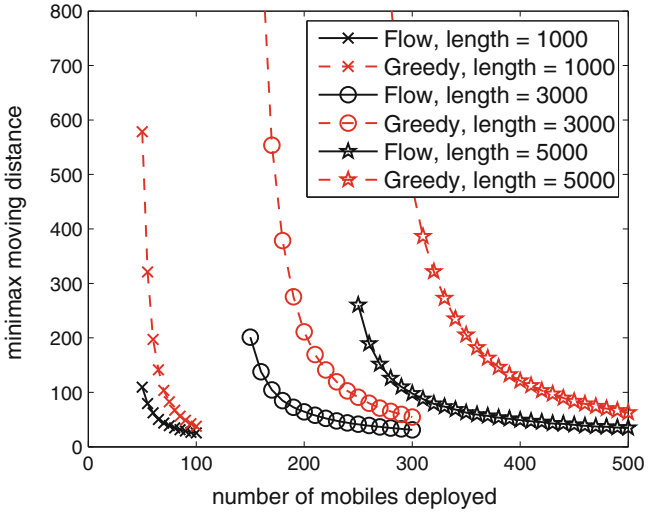


Fig. 21 Minimax moving distance with redundant mobile sensors

5.3 Find and Mend Barrier Gaps

Barrier gaps may occur at deployment if sensors are randomly deployed or in an already formed barrier if some sensors used to form the barrier start malfunctioning or run out of power. In this section we study how to use mobile sensors to improve barrier coverage and investigate its design issues and performance tradeoffs. In particular,

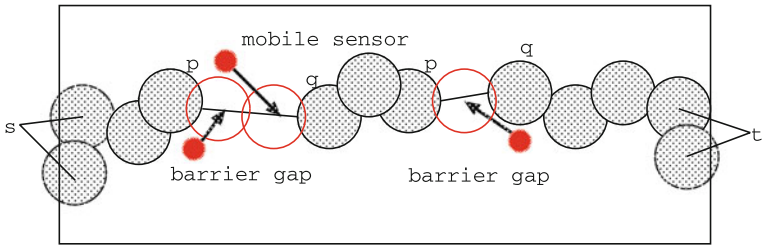


Fig. 22 Illustration of FIND- GAPS algorithm

we would like to find barrier gaps and relocate sensors to fill the gaps and form new barriers. This task faces a number of challenges because of the unique features of barrier coverage and the resource constraints of wireless sensor networks.

First, barrier coverage is a global property, requiring a chain of sensors with overlapping sensing ranges across the entire length of the network. Using mobile sensors to connect local sensor clusters do not necessarily result in the formation of a global barrier. It may require that multiple mobile sensors be relocated to certain desired locations at the same time.

Second, each mobile sensor has a limited moving range due to its energy constraints. Its movement is confined within its moving range and it may not be relocated to arbitrary locations in the network. Moreover, it is highly desirable to balance the energy consumption among mobile sensors to prolong the network lifetime.

We present a two-phase algorithm that can efficiently relocated mobile sensors to form new barriers while balancing the energy consumption among the mobile sensors. In the first phase, the algorithm scans the network from left to right to look for barrier gaps. In the second phase, the algorithm computes which mobile sensors should be relocated to what locations such that the maximum moving distance among all sensors is minimized. This will have the effect of balancing energy consumption. Our approach takes advantage of the underlying line-based sensor deployment and its performance in terms of barrier coverage.

Find Barrier Gaps. Starting from the left boundary, the algorithm greedily looks for a connected cluster of static sensors that extends farthest to the right direction. It then finds the sensor node (marked as q) that is located to the right of the cluster and closest to the rightmost node of the cluster (marked as p). The space between these two sensors is marked as a barrier gap, denoted by (p, q). The process is repeated from sensor q until it reaches the right boundary of the area. This is illustrated in Fig. 22.

Recall that the sensing range of each sensor is denoted by r, and use P to store all the barrier gaps found in the process. The details of the algorithm are described as follows:

FIND- GAPS(N, r)

1. Initialize $P = \emptyset$.
2. Construct a connectivity graph $G(V, E)$ ($V = N \cup s \cup t$) as follows, where each vertex in N represents a static sensor, and s and t are two virtual nodes

representing left and right boundary of the area. $E = \{(u, v), (v, u) | u \in N, v \in N$, and $dist(u, v) <= 2r\} \cup \{(u, s), (s, u) | u \in N$, and dist($u$, left-boundary) $<= r\} \cup \{(u, t), (t, u) | u \in N$, and dist($u$, right-boundary) $<= r\}$.

3. Starting at s, perform a depth-first search for t in G. If successful, the algorithm terminates, and returns P. Otherwise, mark the rightmost node appeared in the search as p, and mark node that is to the right of and closest to x as q. Mark the space between p and q as a barrier gap, and add the gap (p, q) to P.
4. Remove all nodes to the left of q and their associated edges, and set s to q. Repeat from step 3.

After the algorithm terminates, P contains a set of gaps by filling which a new barrier will be formed.

Mend Barrier Gaps with Mobile Sensors. For each gap obtained in the first phase, we relocate mobile sensors to fill the gap and minimize the maximum moving distance among all sensors to balance the energy consumption. When all gaps are mended, a new sensing barrier will be formed.

Consider a gap between node p and node q. The minimum number of sensors needed to fill the gap is $g = \lceil \frac{dist(p,q) - 2r}{2r} \rceil$. Divide the segment (p, q) evenly with g grid points. These grid points represent a set of locations where if each of them is occupied by a mobile sensor, the whole network will be barrier covered. Note that due to energy constraints, each mobile sensor has a certain moving range, and can only move to a location within the range.

Given the set of mobile sensors M, the set of grid points G obtained for all gaps, and the moving range d of mobile sensors, we use a bipartite matching algorithm to compute if every grid point can be occupied by a mobile sensor under the sensor mobility constraint. Should a solution exist, the algorithm also gives the matching between the mobile sensors and the grid points, i.e., an assignment of mobile sensors the grid points. The algorithm is described as follows:

MEND- GAPS(M, G, d)

1. Construct a bipartite graph $G'(V, E)$ $(V = M \cup G)$ as follows, where each vertex in M represents a mobile sensor and each vertex in G represents a grid point in a gap. $E = \{(u, v), (v, u) | u \in M, v \in G$, and $dist(u, v) < d\}$.
2. From $G'(V, E)$, construct a flow graph $G^*(V^*, E^*)$ and assign capacity to each edge as follows: $\forall$ u $\in$ V, add u to V^*; $\forall$ (u, v) $\in E$, add (u, v) to E^*. Set $capacity(u, v) = 1$ for $u \in M$ and $v \in G'$. Add a virtual source node S to V^*, and $\forall$ u $\in M$, add an edge (S, u) to E^*, set $capacity(S, u) = 1$; add a virtual sink node D to V^*, and $\forall$ u $\in G$ of $G'(V, E)$, add an edge (u, D) to E^*, set capacity(u, D) = 1.
3. Use a maximum flow algorithm (e.g., Ford-Fulkerson [36]) to compute and return the maximum flow from S to D in G^*.

The moving range constraint of mobile sensors is accounted for in the algorithm by connecting each mobile sensor only to those grid points within its moving range. When the algorithm terminates, if the returned maximum flow from S to D equals the total number of grid points, each grid point will be assigned a mobile sensor.

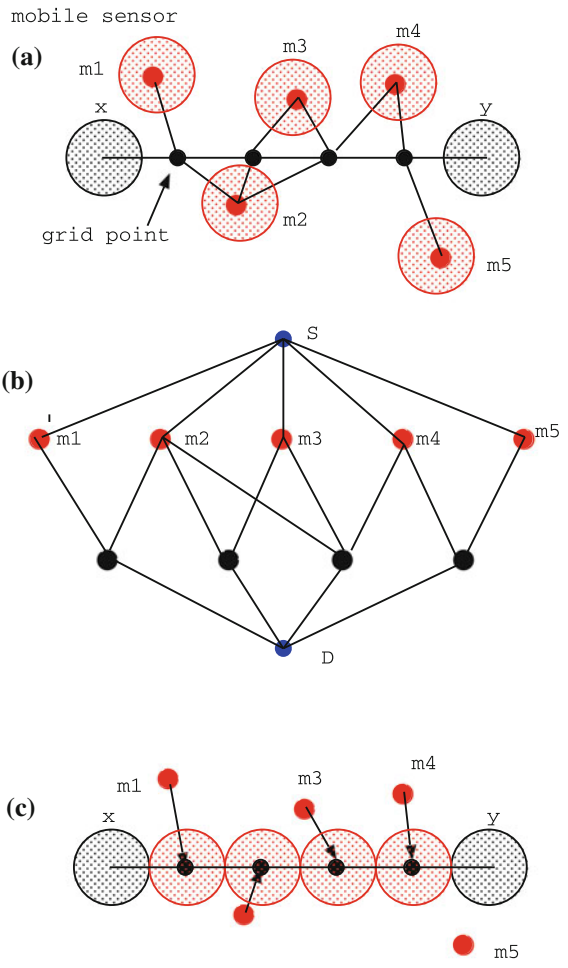


Fig. 23 Illustration of MEND- GAPS algorithm

A new barrier will be formed after the selected mobile sensors move to their assigned grid points. Otherwise, if the returned maximum flow is smaller than the number of grid point, some grid points will not be occupied by sensors, and the gap will not be mended under the mobility constraint. It is straightforward to check that the time complexity of FIND- GAPS is $O(V + E)$, and the time complexity of MEND- GAPS is $O(VE^2)$.

The main steps of the MEND- GAPS algorithm are illustrated in Fig. 23. In part (a), the gap between node p and q is divided evenly by four grid points. Suppose there are five mobile sensors available. We add an edge between a mobile sensor and a grid point if the grid point is within the moving range of the sensor, resulting in a bipartite graph. In part (b), we create a virtual source node S connecting to all mobile nodes,

a virtual sink node D connecting to all grid nodes, and assign unit capacity to each edge. We then use a maximum flow algorithm to compute and return the maximum flow. Based on the results, we relocate appropriate mobile sensors to each grid point to fill the gap, as shown in part (c).

Finally, a binary search-based algorithm can be used to find the minimax moving distance among all sensors that is required to find a bipartite match between mobile sensors and grid points in the gaps.

MINIMAX- MOVING- DISTANCE (M, G)

1 $last\_success \leftarrow \ell$; $last\_fail \leftarrow 0$; $d_{min} \leftarrow \ell$
2 **while** ($last\_success$ - $last\_fail \geq \varepsilon$)
3 **do** $f \leftarrow$ MEND- GAPS(M, G, d_{min})
4 **if** (f = SIZEOF(G))
5 **then** $last\_success \leftarrow d_{min}$
6 **else** $last\_fail \leftarrow d_{min}$
7 $d_{min} \leftarrow (last\_fail + last\_success)/2$
8 return d_{min}

The above binary search-based MINIMAX- MOVING- DISTANCE algorithm terminates in $\Theta(log\ell)$ iterations. When it terminates, it will return the minimum moving distance guaranteeing that every grid point is occupied by a sensor. The ϵ in line 2 of MINIMAX- MOVING- DISTANCE is a termination threshold. It represents the precision of the d obtained from this algorithm.

The performance of the above algorithms are evaluated via simulation. Sensors are deployed along the horizontal central line in a rectangle of size 1000×300. The sensors are deployed on evenly spaced grid points along the line with normally distributed random offsets, as described in Sect. 4.1. All sensors have a sensing range of 10 and three different random offset variances $\sigma = 10$, 30, and 50 are considered. Each data point in this section is an average of 1,000 repeated experiments.

Figure 24 plots the barrier coverage probability as a function of the number of mobile nodes, where the number of stationary sensors $n_s = 100$ and the number of mobile sensors n_m varies from 0 to 100. We focus on the performance of the algorithms without considering the effect of limited moving range of mobile sensors. In other words, we assume that each mobile sensor can be moved to any location in the network. We will later examine the minimum required moving range to achieve barrier coverage.

The barrier coverage probability is zero initially. As more mobile sensors are added to the deployment, the barrier coverage probability starts to increase quickly after a certain point. When the number of mobile sensors is not large enough to fill all the barrier gaps, no new barriers can be formed and the barrier coverage probability remains unchanged. After a certain point, there is a positive probability that the mobile sensors can fill all the gaps to form a barrier, and the probability increases as more mobile sensors are added. When there are enough mobile sensors such that all the gaps can be filled almost surely, the barrier coverage probability reaches one.

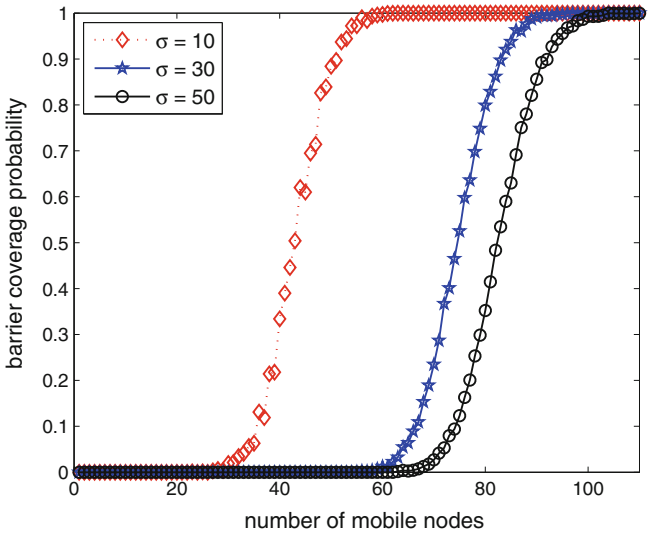


Fig. 24 Barrier coverage probability as a function of the number of mobile sensors

For different deployments with different random offset variances, the larger the variance, i.e., sensors are scattered farther from their target landing points, the more barrier gaps there are and hence more mobile sensors are needed to fill the gaps in order to form a barrier. If the variance $\sigma = 10$, then approximately 45 mobile sensors are needed to form a barrier. For $\sigma = 30$ and 50, achieving barrier coverage would require approximately 75 and 83 mobile sensors, respectively.

The minimum moving range required for mobile sensors to achieve barrier coverage is shown in Fig. 25. As the number of mobile sensors increases, the moving range required to achieve barrier coverage decreases. This demonstrates a tradeoff between the number of mobile sensors and the moving ranges required to achieve barrier coverage. If there are sufficient mobile sensors, mobile sensors may only need to move a small distance to form a barrier. Otherwise, mobile sensors may have to move a larger distance to compensate for the shortage of mobile sensors.

6 Underwater 3-Dimensional Barrier Coverage

Anti-submarine warfare (ASW) is a critical challenge for maintaining a fleet presence in hostile areas. International submarine sales on today's markets are not covered by any nonproliferation treaty, making it possible for nation states, organizations, or even individuals with sufficient resources to purchase submarines. Some of these submarines are capable of launching cruise missiles to deliver conventional, nuclear, chemical, or biological payloads [47]. Drug traffickers have, also, used submarines

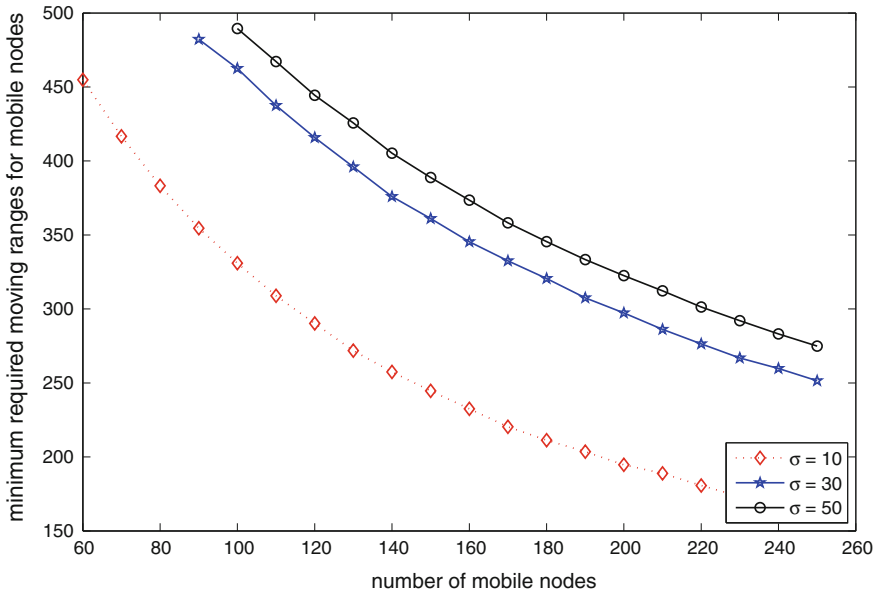


Fig. 25 Minimum required moving range to achieve barrier coverage

to smuggle illegal drugs into the US. The Washington Post reported that a total of thirteen submersible vessels of this kind were captured in 2007 [48], while many more were believed to have evaded US coastal patrols.

To make things worse, recent technology advances have made it possible for submarines to evade standard sonar detection [49]. In particular, submarine hulls can be fitted with rubber anti-SONAR protection tiles to thwart active SONAR detection. These rubber tiles, also, dampen intra-submarine noise to thwart passive acoustic detection at a distance.

Thus, finding alternative ways to detect submarines becomes an important and timely problem. Using magnetic or acoustic sensors in close proximity to the possible underwater pathways of submarines is a viable new approach. Recent advances in micro electro-mechanical systems and communication hardware have made this approach more reliable and relatively inexpensive with a decreased power consumption. Large-scale deployments of underwater wireless sensor networks are expected to become a reality in the near future. These deployments may occur by dispersal from an aircraft or artillery ordinance. Sensors with different buoyancy may submerge to different depths. A sensor node may be equipped with a small ballast tank to control its weight and obtain different buoyancy. Such deployment strategies can deploy many sensors over a vast space quickly, but limit the control of sensor placement.

We want to form a barrier in a sensor network to detect moving objects. In 2-dimensional (2D) terrestrial strip sensor networks, a barrier is a chain of connected sensors from one end of the strip to the other end. No moving objects, regardless

of which paths they choose to pass through, can cross the chain undetected. In 3-dimensional (3D) underwater sensor networks, however, forming a barrier is much more subtle. For example, even if an unbroken chain of overlapping sensor zones between two opposing sides exists in a cuboid, intruders may still be able to pass under or over such a chain without being detected. Thus, an overlapping sensor chain no longer constitutes a barrier in a 3D space. Instead, a barrier in a 3D space should be a set of sensors with overlapping sensing zones of adjacent sensors that covers an entire (curly) surface that cuts across the space.

The goal of our research is to construct a 3D barrier that is scalable for large scale coastline protection in a timely fashion. As underwater movement is very power intensive our solution should minimize the maximal travel distance required of any sensor comprising our barrier. This will maximize the residual energy and thus allow for a longer coverage period.

We present the first set of results for constructing a barrier to detect intruding submarines in a 3D sensor network where sensor nodes are distributed uniformly at random modeled by a Poisson Point Process. We show that a barrier in such a 3D sensor network for a finite density of sensors is unlikely to exist. This suggests the necessity to deploy sensors with at least limited mobility. In order to form an underwater water barrier we plan on moving sensors to set of predefined fixed grid points based on the sensor's detection radius. Such a deployment create a vertical barrier with no holes.

We assign and ultimately move sensors to grid positions based on grid-matching using the Hungarian method, a classic centralized approach. Through simulation we develop a rough lower bound on the maximum travel distance any one sensor needs to travel. This also gives us an upper bound on the computation time. No approximate solution should take longer than an optimal solution.

To reduce the computation cost we consider using a centralized approximate solution based on auction theory. We compare the performance characteristics of the auction-based approach to the optimal approach. We show that for our application auction algorithms provide desirable tradeoffs between computation time and maximum travel distance.

6.1 Network Model

We consider a sensor network consisting of sensors deployed in a large-scale 3D rectangular cuboid. For the initial configuration, we assume that the locations of these sensors are uniformly and independently distributed in the cuboid. Such a random initial deployment is desirable in scenarios where prior knowledge of the region of interest is not available. This may be the result of certain deployment strategies. Under this assumption, the sensor locations can be modeled by a stationary 3D Poisson Point Process. We assume that each sensor can sense the environment and detect intruders in the 3D sphere of radius r.

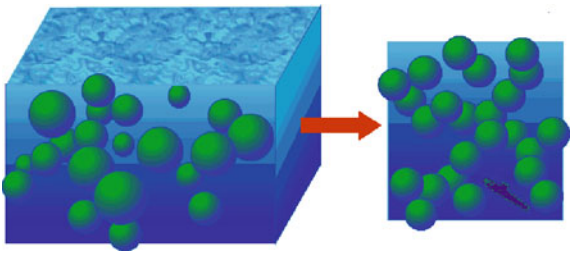


Fig. 26 Barrier coverage in 3D sensor networks

In a 2D sensor network on a strip area, the barrier is directly related to the percolation of the network model [1, 21]. These connected sensor clusters act as "trip wires" to detect any crossing intruders as shown in Fig. 1. In a 3D sensor network, however, a sensor cluster connecting the opposite surfaces of the cuboid no longer constitutes a barrier that can detect crossing intruders. There could be "holes" where an intruder can cross the cuboid undetected. Percolation of sensors no longer provides barrier coverage and intruders can evade detection via the uncovered space. Figure 26 illustrates this effect, where the figure at the right-hand side shows the projection of the 3D cuboid in the direction of the arrow.

We consider a 3D cuboid of size $l \times w \times d$, where l, w, and d denote the length, width, and depth of the cuboid, respectively. A crossing path is a path that connects one surface of the cuboid to the opposite surface, where the ingress point and the egress point reside on two opposite surfaces of the cuboid. A crossing path is said to be covered if it intercepts at least one sensor. Intruders moving along covered crossing paths will be detected. A network provides a barrier if any crossing path attempted by an intruder results in its being detected. A 3D sensor network is said to be barrier covered in a specific direction if any crossing path intersecting the two surfaces perpendicular to the direction goes through the sensing zone of at least one sensor. Without loss of generality, we assume that intruders attempt to penetrate the cuboid in the direction of depth. Clearly, to form a 3D barrier requires a continuous surface that is fully covered by sensors.

6.2 Analytical Results of 3-Dimensional Barriers

The following theorem characterizes the existence of 3D space barriers.

Theorem 4 *In a 3D cuboid of size $l \times w \times d$ where sensors are distributed according to a Poisson Point Process of density λ, no barrier exists in the cuboid in the direction of depth for finite sensor density λ and depth d as $l, w \to \infty$.*

Proof Projecting the 3D cuboid in the direction of the cuboid depth results in a 2D surface of size $l \times w$. After the projection, a "barrier tunnel" in the original 3D cuboid where an intruder can pass through may not be present in the projected 2D surface.

Therefore, the barrier coverage of the 3D cuboid is bounded above by the barrier coverage of its projected 2D surface. If an intruder can penetrate the projected 2D surface undetected then there exists for sure an uncovered crossing path in the 3D cuboid.

In the asymptotic case, when $l, w \to \infty$, the sensors on the projected 2D surface follow the Poisson point process distribution of density λd, with each point occupied by a disk of radius r. The fraction of the area that is covered in the projected 2D surface, f_a, is obtained by [21, Theorem 1]. That is,

$$f_a = 1 - e^{-\lambda d \pi r^2} < 1 \text{ if } \lambda, d < \infty.$$

Therefore, there is uncovered area in the projected 2D surface for finite sensor density and cuboid depth. An uncovered area in the project 2D surface corresponds to a "barrier tunnel" in the original 3D space where an intruder can pass through undetected. As a result, there is no barrier in the original 3D cuboid in the direction of depth. □

6.3 Stealth Distance in 3D Sensor Networks

In a wireless sensor network, as it moves across the network, an intruder will be detected whenever its path intersects with the sensing sphere of a sensor. It is interesting to investigate the distance an intruder travels before first being detected by sensors. This distance, referred to as the *stealth distance* and defined below, measures the intrusion detection performance of the sensor network (Fig. 27).

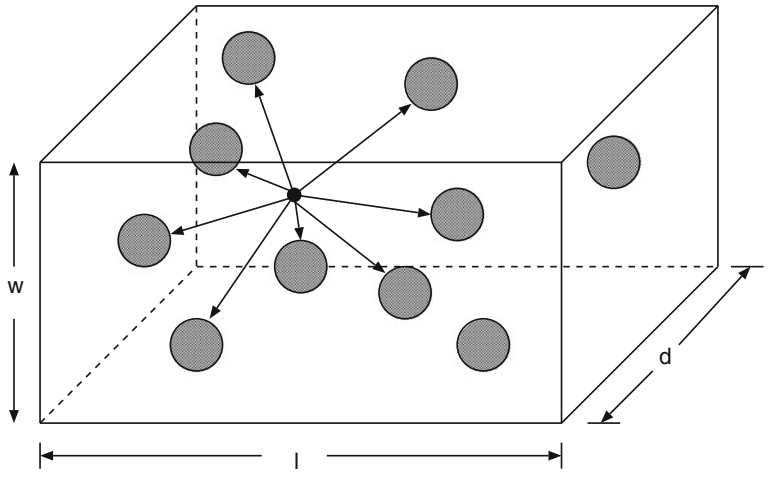


Fig. 27 Stealth distance of intruder in 3D sensor networks

Definition 2 *Consider a 3D sensor network where sensors of sensing range r are distributed according to a Poisson point process of density λ. Assuming that an intruder is initially undetected and moves in a random direction along a straight line. The stealth distance of the intruder, X, is defined as the distance it travels before it is first detected by any sensor.*

The notion of stealth distance was introduced in [50], and the authors derived an approximation of the expected stealth distance for 2D sensor networks. In the following theorem, we characterize the distribution of the stealth distance in a 3D sensor network. The distribution of the stealth distance in a 2D sensor network can be obtained similarly.

Theorem 5 *In a 3D sensor network where sensors are distributed according to a Poisson point process of density λ, the stealth distance of an intruder, X, follows an exponential distribution with parameter $\lambda\pi r^2$, i.e.,*

$$\mathbb{P}(X < x) = 1 - e^{-\lambda\pi r^2 x}. \tag{10}$$

The full details of this proof can be found in [9]. It follows from Theorem 5 that the expected stealth distance of a randomly located intruder is

$$E[X] = \frac{1}{\lambda\pi r^2},$$

which is inversely proportional to the sensor density (λ) and the projected area of the sensing sphere (πr^2). Therefore, in order to shorten the stealth distance of an intruder, one can add more sensors, or use sensors with larger sensing range. To guarantee that the expected stealth distance of an intruder be smaller than a specific threshold l_0, we should have

$$\frac{1}{\lambda\pi r^2} < l_0.$$

The above relationship between the stealth distance and the deployment parameters (sensor density and sensing range) provides important guidelines to the planning of sensor networks for intrusion detection.

6.4 Barrier Construction

The first step in constructing our 3D barrier is the development of some basic intuitions about the nature of the problem. Our first question is "What is the time complexity needed to construct the optimal assignment for varying sizes of sensor fields?" The optimal solution will give an upper bound on the time complexity allowed to construct approximate solutions. Our second question is "What is the maximum travel distance required of any sensor in the optimal solution and how does it change

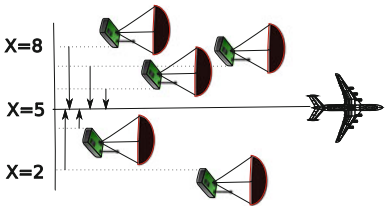


Fig. 28 Phase one: an air drop of sensors along a path resulting in scattered placement along that path

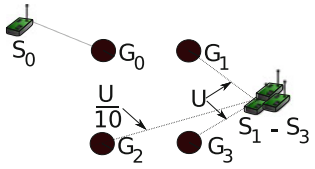


Fig. 29 Phase two: sensors $S_1 - S_3$ derive the same utility U, from being assigned to G_1 or G_3. All receive a minute utility $(.1U)$, from being assigned to G_2. S_0 does not have the energy to move to any grid point other than S_0. An auction occurs to assign $S_1 - S_3$ to $G_1 - G_3$

for varying sizes of sensor fields?" This allows us to understand the movement distance requirements that must be placed on sensors during deployments. It also allows us to compare approximate solutions to the optimal solution. This section analyzes the optimal solution and offers an alternative approximate solution that preserves the maximal travel distance while offering a considerable time complexity improvement.

The results in the previous subsection show that barriers in a 3D network are unlikely to exist with Poisson distributed stationary emplacement sensors. To overcome this obstacle, we propose to use mobile sensors to form a barrier. To accomplish this task, we require that sensors have the following features: (1) *X, Y, Z positional control:* be able to move across the surface of the ocean and also to different depths; *Localization:* be able to identify their own X, Y, Z position to some degree of certainty. Note that sensors may be able to localize their own positions, after their deployment, using a set of anchor nodes. Each of the anchor nodes is equipped with an underwater acoustic modem that submerges in the water, and a GPS that stays on the water surface. The GPS provides the location information of the node. An underwater sensor may communicate with a set of anchor nodes through their underwater acoustic modems and use the propagation delays of acoustic signals and the locations of the anchor nodes to calculate its location.

We propose a three-phased approach for constructing a 3D barrier, which minimizes the maximum energy used by any sensor:

1. Find an optimal vertical plane at which the sensors will form a grid-based barrier as shown in Fig. 28.
2. For each sensor, identify its best grid-based assignment as shown in Fig. 29.

3. Move sensors from their initial dropped position directly to their final assigned grid positions.

Finding the optimal barrier location. Suppose as in Fig. 28 an airplane drops a set of mobile sensors such that plane X is parallel to the coast line to be protected. The sensors must move to some plane X, such that X minimizes the energy expended by any one sensor. For this calculation, we are assuming that the sensors will only be moving in one direction to approach the X location.

Suppose X represents the optimal line for each of the m sensors to approach. Let $d(x_i, x)$ be $|x - x_i|$ the distance traveled by any sensor s_i in the direction of X. We are attempting to minimize the maximal distance traveled by any sensor. To accomplish this let $x = (\max x_i - \min x_i)/2$, then the maximum distance any one sensor travels is minimized. In Fig. 28, all the sensors will move to the line $X = 5$. It is straightforward to show that X must be somewhere inside the set x-coordinates of the sensors. As any two sensors move to meet one another, the total distance they travel is the distance between them. Thus, the sensors on either edge move the farthest, and half the distance between them minimizes the maximum distance any one sensor moves.

Optimal assignment of sensors to grid points. Assigning n sensors to n grid positions is related to the Assignment Problem, where we would seek a one-to-one matching between sensors and grid positions. The cost of each possible matching is the energy required to move the sensor to a grid point. A classic optimal solution is the Hungarian Method and can be computed in $O(n^3)$, where n is the number of sensors [51]. It produces a one-to-one assignment which minimizes the total energy expended.

However, in our context, we are looking to minimize the maximal amount of energy drain any one sensor incurs moving to a grid position. This will maximize the network lifetime. Let k be the maximal travel distance we allow any sensor to travel for moving to cover a grid position. We seek to find the smallest value of k such that each sensor expends at most k, yet all sensors can move to cover a grid position.

We can apply the Hungarian Method to compute k. Assume we have a function $HungarianK(S, G, k) \rightarrow Assignment, k'$. This function takes a list of sensor positions S, a list of grid positions G which are to be covered, a maximal value k, and computes a feasible assignment. If all the edges in the intermediate result do not exceed k then the assignment is returned otherwise the empty-set is returned. In addition k' represents the largest edge found in the assignment.

With this new function a centralized solution can be accomplished at a central node as follows. Sensors transmit positional information to a central node. The central node creates a sorted list of edge weights. Each sensor can be paired with every grid position resulting in a list of edge weights whose number is n^2. A binary search is conducted on the edge list and the midpoints of the edge weight list are fed as the parameter k to $HungarianK$. The detailed algorithm description can be found in [9]. The running time for a solution will be $O(n^3 \log n)$, where $O(n^3)$ is for each running of Hungarian Method and $O(\log n)$ for the binary search.

Auction-based assignment of sensors to grid points. The classic Hungarian method provides a centralized optimal solution to our assignment problem. To reduce computation cost, we present a centralized approximate solution to the problem of assigning sensors to grid positions using *auction algorithms* [52]. Auction algorithms offer a reduced computational expense to the Hungarian method, while delivering close approximations to the optimal solution.

Auction algorithms arrive at an approximately optimal solution based on a set of agents bidding for the same resource, where each sensor labeled as i is bidding to be assigned to a particular grid position labeled as j. Each sensor associates a utility, which in our case is inversely proportional to a sensor's distance to a grid point. Let u_{ij} denote the utility of assigning sensor i to grid point j. Closer grid points have a higher utility to a sensor. Thus a sensor is willing to pay an increased price to compete for a grid point that maximizes its utility. The global known price for grid point j is denoted by p_j. The value $u_{ij} - p_j$ derives the benefit a sensor i achieves by being assigned to grid point j. Therefore, if the price at a particular grid point increases then a sensor will opt for a grid point with a lower utility and a higher benefit if it exists.

The algorithm starts by constructing an arbitrary assignment of sensors to grid positions. If each sensor derives the maximum benefit from its current assignment then the algorithm terminates. Otherwise, a sensor i seeks to be associated with j_i, which maximizes its benefit as shown in Eq. (11).

Once the optimal candidate is identified, i raises p_j by the amount of $Increase$, which is the difference between the best benefit and the next best benefit. Now sensor i is associated with j_i and whoever j_i was associated with is associated with j. As j_i has the optimal benefit, $BestBen \geq NextBestBen$, therefore $Increase \geq 0$.

$$u_{ij_i} - p_{j_i} = \max_{j=1,\ldots,n} (u_{ij_i} - p_{j_i}) \tag{11}$$

$$p_{j_i} = p_{j_i} + Increase \tag{12}$$

$$Increase = BestBen - NextBestBen \tag{13}$$

$$BestBen = \max_{j}(u_{ij} - p_{j_i}) \tag{14}$$

$$NextBestBen = \max_{j \neq j_i}(u_{ij} - p_{j_i}) \tag{15}$$

It could turn out that $BestBen = NextBestBen$, i.e., a sensor derives the same benefit from being assigned to either of two separate grid positions. In that case, the bid increment would be zero. This could lead to repeated exchanges among a small group of sensors without increasing the price. This would result and an infinite loop and the system would fail to converge. To force convergence, we require that any bid must increase the price. To accomplish this in the algorithm, we replace Eq. (13) with Eq. (17) with a positive value ε. Further, we say that the system is in ε-equilibrium

Table 1 An example of a situation where a naive auction can lead to bids with no price increase

Round	Prices	Mappings	Bidder	Pref.	Inc.
1.	(0,0,0)	$S_1 : G_1, S_2 : G_2, S_3 : G_3$	2	1	0
2.	(0,0,0)	$S_1 : G_2, S_2 : G_1, S_3 : G_3$	1	1	0
3.	(0,0,0)	$S_1 : G_1, S_2 : G_2, S_3 : G_3$	2	1	0

if each sensor is no more than ε off the maximum benefit possible. We accomplish this by substituting Eq. (11) with Eq. (16).

$$u_{ij_i} - p_{j_i} >= \max_{j=1,\ldots n} (u_{ij_i} - p_{j_i}) - \varepsilon \tag{16}$$

$$Increase = BestBen - NextBestBen + \varepsilon \tag{17}$$

We present an example to show how infinite loops might occur during the bidding process in phase two based on Eqs. (11) and (13). Suppose that the sensors have limited movement so that S_0 can only bid for position G_0. Suppose that $S_1 - S_3$ are equidistant from G_1 and G_3. Then the sensors derive the same utility, denoted by U, from being assigned to G_1 or G_3. Suppose they can move all the way to G_2, but it would drain their energy reserves almost completely. Thus, G_2 offers a small nonzero utility to the sensors, which we denote by $(0.1 \times U)$. Finally, suppose the initial mapping is $S_0 \rightarrow G_0, S_1 \rightarrow G_1, S_2 \rightarrow G_2, S_3 \rightarrow G_3$. To keep this example as simple as possible we remove S_0 and G_0 from our example. This configuration is depicted in Fig. 29. The resulting infinite loop is presented in Table 1. The top two benefits are identical for all the sensors and the initial price for all grid points is zero. Therefore, sensors S_1, S_2, and S_3 will wind up endlessly swapping assignments.

If we substitute in Eqs. (16) and (17) we see the prices associated with being assigned to a grid point increasing. This is depicted in Table 2. During each round of bidding the price for either G_1 or G_3 is increased by at least ε. Let $m\varepsilon$ be the first price that exceeds the utility U. When such a price is reached for either grid point, the benefit $(U - m\varepsilon) < 0$. At this point, the small utility $(0.1 \times U)$, offered by G_2 to any sensor and its zero price will cause a sensor to prefer to be assigned to G_2. Once a sensor chooses G_2, we will have a feasible assignment and the algorithm can terminate. As demonstrated, if a feasible assignment exists then this algorithm is guaranteed to terminate in a finite number of steps (see [52] for details). Once completed, sensors move from their initial positions to their assigned grid positions.

Table 2 Example of increasing bids for a grid position

Round	Prices	Mappings	Bidder	Pref.	Inc.
1.	(0,0,0)	$S_1 : G_1, S_2 : G_2, S_3 : G_3$	2	1	ε
2.	(ε,0,0)	$S_1 : G_2, S_2 : G_1, S_3 : G_3$	1	3	2ε
3.	(ε,0,2ε)	$S_1 : G_3, S_2 : G_2, S_3 : G_2$	2	1	2ε
4.	(3ε,0,2ε)	$S_1 : G_2, S_2 : G_1, S_3 : G_2$	1	2	2ε
...	...	...	...	...	...

Eventually the price will exceed the utility afforded by an assignment and a sensor bid for grid position G_2, allowing the algorithm to terminate

6.5 Performance Evaluation

We use simulations to evaluate how the approximate solution performs under our assumed deployment strategy and give some comparisons to the optimal solution. The average ocean depth is 3,790 m [53]. Specialized submarines such as the Trieste can dive to 11,015 m in the water [54]. Other mainstream submarines such as the Komsomolets series have dived to 1,300 m [55]. Our simulations propose a sensor network that would be able to detect submarines up to 3,790 m.

Recent studies have stated that commercial motes with magnetometers can detect submarines at distances of several hundred meters. For our simulation, we chose a fixed detection range of 460 m and if sensors are deployed every 640 m in a grid pattern there will be no uncovered space. A column of six sensors with the first being deployed at a depth of 320 m allows coverage of 3840 m. We allow for our sensors to sink from between 320 and 3,520 m. The resulting cube's *depth* is 3,200 m. We allow sensors to drift away from the drop position with a radius up to 160 m in any direction, so a cube's *width* is 320 m. Finally, each sensor column is 640 m apart. We assume that the aircraft uniformly drops sensors from the first column to the final column. The *length* of a cube ranges from 1,920 to 55,680 m. Our simulations were run on an Intel Xeon™ 3.20 GHz CPU. The operating system was Red Hat Fedora Core 8. All software was written in Python.

Performance of the Centralized Optimal Solution. First, we compare the running time of the traditional $Hungarian$ Method and our variant $HungarianK$ for different sized cubes. The former computes the minimum total weight solution and the latter computes the minimum maximum edge weight.

Figure 30 depicts our results. The x-axis represents the number of sensor columns being tested, which each sensor column being comprised of six sensors. The y-axis represents the total computation time required to achieve a result. This graphic shows that the computation time for computing HungarianK is within $O(n^3 \log n)$.

Figure 31 depicts the maximum movement any one sensor must travel to get to its optimal assignment. The y-axis represents the maximum distance any one sensor must travel. As the network size increases, the $Hungarian$ Method requires more movement from sensors. Figures 30 and 31 show that there is a tradeoff between the computation time/energy and the total movement distance. This is an important

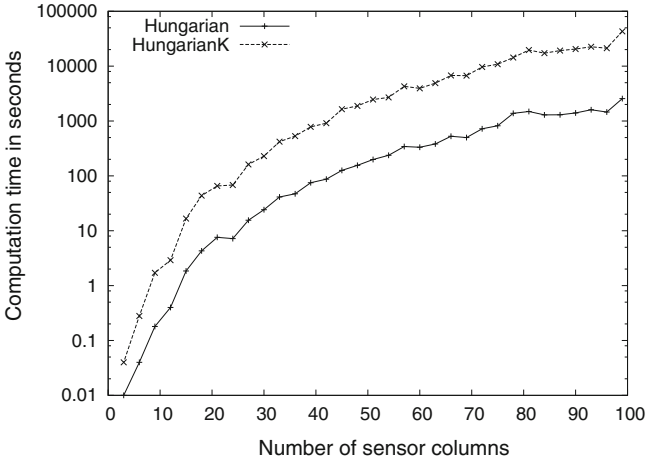


Fig. 30 Comparison of running time between the $Hungarian$ and our $HungarianK$ variant

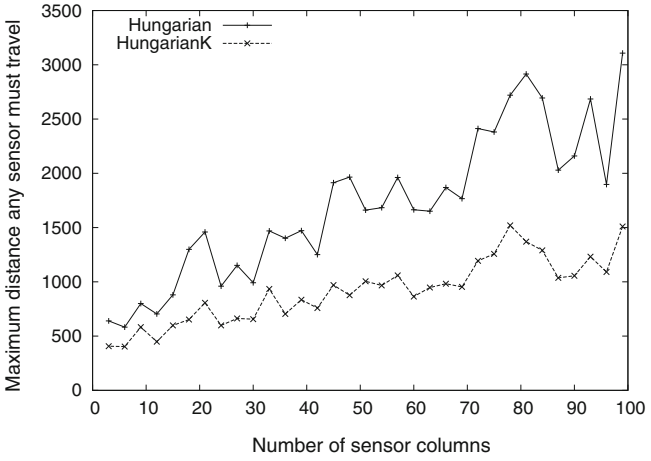


Fig. 31 Comparing the $Hungarian$ Method with our $HungarianK$ variant to determine maximum moving distance

consideration in energy constrained environments where movement consumes far more energy than computation.

Performance of the Centralized Approximate Solution. To seed the initial assignment during each run, we associated each sensor to an arbitrary grid position for all but the last experiment. This mapping was chosen to allow for an unbiased estimate of how quickly the system would converge. After some initial analysis, we chose five different values of ε to show how the system behaved. The values chosen were 0.5, 1.0, 2.0, 5.0, and 10.0. Each experiment computed a result using a single value of ε.

In the last experiment we started the search for a solution with $\varepsilon = 10.0$. Once a feasible solution was determined, the solution was used as an initial assignment to find a solution with $\varepsilon = 5.0$. Any of previous assignments who satisfy the reduced maximal travel distance imposed can be used as part of the solution.

Those re-used assignments may not be optimal, however as they are acceptable we can use them as a partial solution. This reduces the remaining assignments to be computed and therefore reduces the overall time complexity. This search strategy can be considered as a *simulated annealing* process [56]. We continued to reduce the ε value until $\varepsilon = 0.5$. Our simulations show how varying values of ε impact the optimal match between sensors and grid positions. We discuss different tradeoffs arising from varying the values of ε. The annealed-search-based auction is denoted in the figures by the lines $\varepsilon = Var$.

In Fig. 32, the x-axis represents the number of columns in the network with 6 sensors per column. The y-axis represents the average maximum number of bids any sensor makes to be associated with the grid position of its choosing. We can see that smaller values of ε result in a far higher number of bids. The annealed search results in generating numbers of bids similar to the smallest value of ε. For simplicity, this simulation assumes that all sensors communicate in a synchronous fashion with no communications loss.

In Fig. 33, the x-axis is the same and the y-axis represents the total number of bids placed in the average case of the bidding process. As the network size increases, the total number of bids for smaller values of ε increase very quickly. If small values of ε are chosen in a real-world deployment, the large volume of messages required to be transmitted and received would have a significant impact on battery lifetime. This would result in a tradeoff of sensor lifetime versus coverage.

In Fig. 34, the x-axis is the same and the y-axis represents the maximum number of meters any sensor must travel. The smaller the value of ε, the shorter the travel

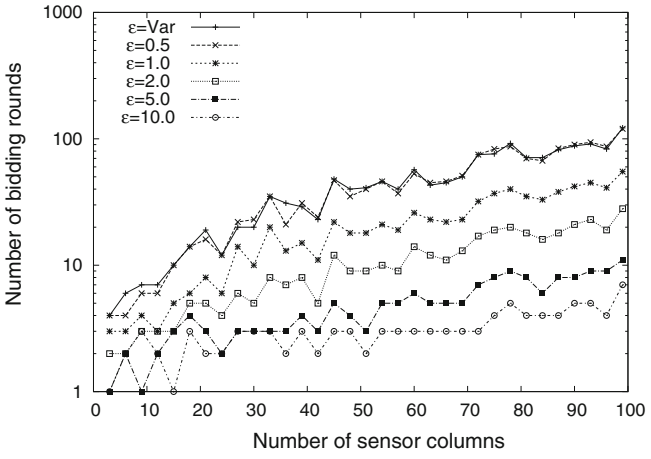


Fig. 32 Size versus number of rounds for varying values for ε

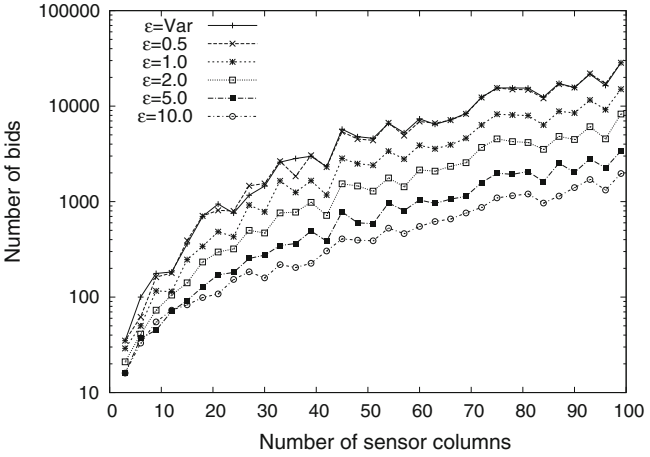


Fig. 33 Network size versus the average total number of bids broadcast for associating all sensors to grid positions given varying values for ε

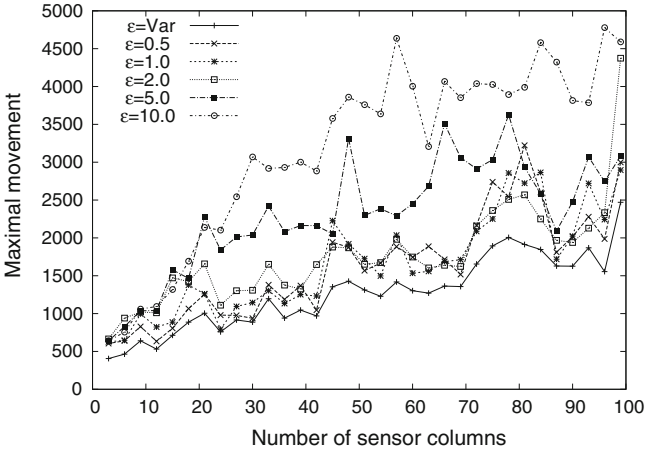


Fig. 34 Network size versus maximal movement required of individual sensor when varying ε

distance for any sensor. In this case, the annealed search outperforms all others. On average, the Hungarian method requires 1.34 times the distance of the annealed search and the modified Hungarian method with binary search, denoted by HungarianK, is 0.75 times better than the annealed search.

In Fig. 35, the x-axis is the same and the y-axis represents the computation time of the approximate solution. For the annealed search, we see that the computation is more expensive when the number of sensor columns is smaller. As the number of sensor columns increase, the annealed search offers a noticeablecomputational

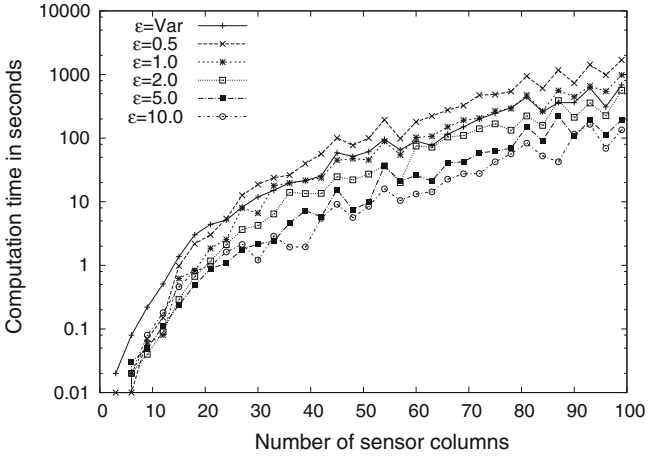


Fig. 35 Network size versus computation time for the different values of ε

improvement over smaller values of ε. In general, the computation of the Hungarian method is comparable to when $\varepsilon = 0.5$. The HungarianK function grows $O(\log n)$ faster making it difficult to simultaneously display all curves in a single graph.

7 Summary

In this chapter we provide a comprehensive survey on the barrier coverage of wireless sensor networks. In particular, we discuss how the barrier coverage of a wireless sensor network depends on the various factors, including the network environments, sensor deployment strategies, and sensor mobility. For the various network scenarios, we present the challenges of providing barrier coverage under the specific network settings, establish the fundamental performance limits, and propose effective and efficient solutions to the construction of sensor barriers. The results should provide important insights into the deployment, design, and performance for wireless sensor network applications for barrier coverage. The analytical techniques and algorithms presented will also shed light on the future research on this topic.

Acknowledgments This research was supported in parts by the National Science Foundation under grants CNS-0953620 and CNS-1018303.

Appendix

Proof of Theorem 2

Proof Consider the relative positions of nodes s_i and s_{i+1}. Define $z^x = s^x_{i+1} - s^x_i$ and $z^y = s^y_{i+1} - s^y_i$. Then $z^x = \zeta + \delta^x_{i+1} - \delta^x_i$ and $z^y = \delta^y_{i+1} - \delta^y_i$. Since the δ^s are Normally distributed with variance σ^2, then $\delta^x_{i+1} - \delta^x_i$ and $\delta^y_{i+1} - \delta^y_i$ are both Normal random variable with variance $2\sigma^2$, i.e., $z^x \sim N(\zeta, 2\sigma^2)$ and $z^y \sim N(0, 2\sigma^2)$.

The distance between s_i and s_{i+1} is equal to:

$$d_i \triangleq |s_{i+1} - s_i| = \sqrt{(z^x)^2 + (z^y)^2} \tag{A.1}$$

This implies that d_i follows a Ricean distribution. Since the distribution is identical for all i, we drop the index and denote the distance between two consecutive nodes as d.

In particular, the probability that $d < \rho$ is given by (see for instance [57], Chap. 2):

$$P(d < \rho) = 1 - Q_1(\frac{\zeta}{\sqrt{2}\sigma}, \frac{\rho}{\sqrt{2}\sigma}) \tag{A.2}$$

where Q_1 is the Marcum's Q-function of the first order, defined by:

$$Q_1(\frac{\zeta}{\sqrt{2}\sigma}, \frac{\rho}{\sqrt{2}\sigma}) = e^{-(\zeta^2+\rho^2)/4\sigma^2} \sum_{k=0}^{\infty} \left(\frac{\zeta}{\rho}\right)^k I_k\left(\frac{\zeta\rho}{2\sigma^2}\right) \tag{A.3}$$

and I_k is the kth order modified Bessel function of the first kind.

Thus, two sensors s_i and s_{i+1} provide barrier coverage with probability $P(d < \rho)$. If each pair of sensors s_i and s_{i+1} is within ρ of each other and within ρ of the boundary, then the sensor deployment provides barrier coverage over the all width of the area. For all $1 \leq i \leq n-1$, denote by W_i the event that $d_i < \rho$. Denote by W^b_0 the event that s_1 is within distance ρ of the boundary $x = 0$, and W^b_n that s_n is within distance ρ of the boundary $x = l$. Since this is not the only configuration that provides barrier coverage, we have

$$P(\text{Barrier Coverage}) \geq P\left(W^b_0 \bigcap W^b_n \bigcap (\bigcap_{i=1}^{n-1} W_i)\right). \tag{A.4}$$

Assumption (1) allows us to consider W^b_0, W^b_n and W_i, $1 \leq i \leq n-1$, as independent events and approximate $P(W^b_0 \bigcap W^b_n \bigcap (\bigcap_{i=1}^{n-1} W_i))$ with $P(W^b_0)P(W^b_n)(P(d < \rho))^{(n-1)}$. Indeed, if assumption (1) was violated, and ρ was almost equal to ζ, then a perturbation which brings node s_i close to s_{i-1} would also create a gap between s_i and s_{i+1}. W_i happening thus implies that W_{i+1} would not happen, and that

both events are conditioned on each other, not independent. However, by choosing the right parameter κ, the approximation by independent events is appropriate, as we confirm in the evaluation section.

We can also easily verify that $P(W_0^b) > P(d < \rho)$ and symmetrically, $P(W_n^b) > P(d < \rho)$, so that $P(\text{Barrier Coverage}) \geq P(d < \rho)^{n+1}$.

Assumption (2) ensures that the gap between $P(BarrierCoverage)$ and $P(\bigcap_{i=0}^{n} W_i)$ stays limited, and that the most likely configuration to provide coverage is indeed by having each s_i and s_{i+1} within ρ of each other. Other configurations are possible, and a gap between s_i and s_{i+1} could be filled by having a third sensor out of position in the sequential ordering along the x-axis. However, assumption (2) ensures that such other configurations have a low likelihood. Simulation will show that, under assumption (2) the lower bound is actually tight.

Note that we do not put an explicit dependency of ζ on n, but as $n \to \infty$, the probability of barrier coverage goes to 1 as $\zeta \to 0$, all other parameters being constant. □

References

1. B. Liu, O. Dousse, J. Wang, A. Saipulla, Strong barrier coverage of wireless sensor networks, in *Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)* (2008)
2. U. Berkeley, M. Co., Tracking vehicles with a uav-delivered sensor network. http://www-bsac.eecs.berkeley.edu/~pister/29Palms0103/
3. A. Saipulla, B. Liu, J. Wang, Barrier coverage with airdropped sensors, in *Proceedings of IEEE MilCom* (2008)
4. A. Saipulla, C. Westphal, B. Liu, J. Wang, Barrier coverage of line-based deployed wireless sensor networks, in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)* (2009)
5. K. Dantu, M. Rahimi, H. Shah, S. Babel, A. Dhariwal, G.S. Sukhatme, Robomote: enabling mobility in sensor networks, in *Proceedings of the IEEE International Conference in Infomation Processing in Sensor Network (IPSN)* (2005)
6. Khepera robots. http://www.k-team.com
7. A. Saipulla, B. Liu, G. Xing, X. Fu, J. Wang, Barrier coverage with sensors of limited mobility. in *Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)* (2010)
8. A. Saipulla, B. Liu, J. Wang, Finding and mending barrier gaps in wireless sensor networks, in *Proceedings of IEEE Globecom* (2010)
9. S. Barr, J. Wang, B. Liu, An efficient method for constructing underwater sensor barriers. J. Commun. **6**(5), 370–383 (2011)
10. S. Meguerdichian, F. Koushanfar, M. Potkonjak, M.B. Srivastava, Coverage problems in wireless ad-hoc sensor networks, in *Proceedings of IEEE Infocom*, pp. 1380–1387 (2001)
11. K. Chakrabarty, S.S. Lyengar, H. Qi, E. Cho, Grid coverage for surveillance and target location in distributed sensor networks. IEEE Trans. Comput. **51**, 1448–1453 (2002)
12. S. Shakkottai, R. Srikant, N. Shroff, Unreliable sensor grids: coverage, connectivity and diameter, in *Proceedings of IEEE Infocom* (2003)
13. X.Y. Li, P.J. Wan, O. Frieder, Coverage in wireless ad-hoc sensor networks. IEEE Trans. Comput. **52**(6), 753–763 (2003)

14. Y. Zou, K. Chakrabarty, Uncertainty-aware and coverage-oriented deployment for sensor networks. J. Parallel Distrib. Comput. **64**, 788–798 (2004)
15. M. Cardei, J. Wu, Energy-efficient coverage problems in wireless ad hoc sensor networks. J. Comput. Commun. **29**(4), 413–420 (2006)
16. Y. Zou, K. Chakrabarty, Distributed mobility management for target tracking in mobile sensor networks. IEEE Trans. Mobile Comput. (TMC) **6**(8), 872–887 (2007)
17. H. Zhang, J. Hou, Maintaining sensing coverage and connectivity in sensor networks, in *Proceedings of the Invited Paper in International Workshop on Theoretical and Algorithmic Aspects of Sensor, Ad Hoc Wireless and Peer-to-Peer, Networks* (2004)
18. G. Xing, X. Wang, Y. Zhang, C. Lu, R. Pless, C.D. Gill, Integrated coverage and connectivity configuration for energy conservation in sensor networks. ACM Trans. Sens. Netw. **1**(1), 36–72 (2005)
19. H.M. Ammari, S.K. Das, Integrated coverage and connectivity in wireless sensor networks: a two-dimensional percolation problem. ACM Trans. Comput. **57**(10), 1423–1434 (2008)
20. D. Gage, Command control for many-robot systems, in *Proceedings of the Nineteenth Annual AUVS Technical Symposium (AUVS-92)* (1992)
21. B. Liu, D. Towsley, A study on the coverage of large-scale sensor networks, in *The 1st IEEE International Conference on Mobile Ad-hoc and Sensor Systems* (2004)
22. C. Shen, W. Cheng, X. Liao, S. Peng, Barrier coverage with mobile sensors, in *Proceedings of IEEE the International Symposium on Parallel Architectures, Algorithms, and Networks (ISPAN)* (2008)
23. S. He, J. Chen, X. Li, S. Shen, Y. Sun, Cost effective barrier coverage by mobile sensor networks, in *Proceedings of IEEE Infocom* (2012)
24. L. Kong, Y. Zhu, M.Y. Wu, W. Shu, Mobile barrier coverage for dynamic objects in wireless sensor networks, in *Proceedings of IEEE MASS* (2012)
25. S. Kumar, T.H. Lai, A. Arora, Barrier coverage with wireless sensors, in *Proceedings of ACM Mobicom* (2005)
26. G. Yang, D. Qiao, Barrier information coverage with wireless sensors, in *Proceedings of IEEE Infocom* (2009)
27. G. Yang, D. Qiao, Multi-round sensor deployment for guaranteed barrier coverage, in *Proceedings of IEEE Infocom* (2010)
28. Y. Wang, G. Cao, Barrier coverage in camera sensor networks, in *Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)* (2011)
29. H. Ma, M. Yang, D. Li, Minimum camera barrier coverage in wireless camera sensor networks, in *Proceedings of IEEE Infocom* (2012)
30. A. Chen, S. Kumar, T.H. Lai, Designing localized algorithms for barrier coverage. in *Proceedings of ACM Mobicom* (2007)
31. A. Chen, T.H. Lai, D. Xuan, Measuring and guaranteeing quality of barrier-coverage in wireless sensor networks, in *Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)* (2008)
32. L. Zhang, J. Tang, W. Zhang, Strong barrier coverage with directional sensors, in *Proceedings of IEEE Globecom* (2009)
33. M. Franceschetti, O. Dousse, D. Tse, P. Thiran, Closing the gap in the capacity of random wireless networks, in *Proceedings of Information Theory Symposium (ISIT)* (2004)
34. G.R. Grimmett, *Percolation* (Springer, Heidelberg, 1999)
35. A. Schrijver, *Combinatorial Optimization* (Springer, Heidelberg, 2003)
36. T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, *Introduction to Algorithms*, 2nd edn. (MIT Press and McGraw-Hill, Cambridge, 2001)
37. Crossbow Technology INC. http://www.xbow.com/
38. P. Balister, B. Bollobas, A. Sarkar, S. Kumar, Reliable density estimates for coverage and connectivity in thin strips of finite length, in *Proceedings of ACM Mobicom* (2007)
39. E. Gilbert, Random plane networks. J. SIAM **9**, 533–543 (1961)
40. A.A. Somasundara, A. Ramamoorthy, M.B. Srivastava, Mobile element scheduling with dynamic deadlines. IEEE Trans. Mobile Comput. (TMC) **6**(4), 1142–1157 (2007)

41. A. Howard, M. Mataric, G. Sukhatme, Mobile sensor network deployment using potential fields: a distributed, scalable solution to the area coverage problem, in *DARS 02* (2002)
42. Y. Zou, K. Chakrabarty, Sensor deployment and target localization based on virtual forces, in *Proceedings of IEEE Infocom* (2003)
43. G. Wang, G. Cao, T.L. Porta, Movement-assisted sensor deployment, in *Proceedings IEEE Infocom* (2004)
44. S. Chellappan, W. Gu, X. Bai, D. Xuan, B. Ma, K. Zhang, Deploying wireless sensor networks under limited mobility constraints. IEEE Trans. Mobile Comput. (TMC) **6**(10) 1142–1157 (2007)
45. B. Liu, P. Brass, O. Dousse, P. Nain, D. Towsley, Mobility improves coverage of sensor networks, in *Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)* (2005)
46. P. Shor, J. Yukich, Minimax grid matching and empirical measures. Ann. Probab. **19**(3), 1338–1348 (1991)
47. Global submarine proliferation: emerging trends and problems (2008). http://www.nti.org/e_research/e3_74.html
48. J. Forero, Drug traffic beneath the waves. In: Washington Post, p. A04 (February 6, 2008)
49. J. Kong, J. Hong Cui, D. Wu, M. Gerla, Building underwater ad-hoc networks and sensor networks for large scale real-time aquatic applications, in *Proceedings of IEEE Military Communications Conference (MILCOM)* (2005)
50. C. Gui, P. Mohapatra, Power conservation and quality of surveillance in target tracking sensor networks, in *Proceedings of ACM Mobicom* (2004)
51. H. Kühn, The Hungarian Method for the Assignment Problem. Naval Res. Logistics Q. **2**, 83–97 (1955)
52. D. Bertsekas, The auction algorithm: a distributed relaxation method for the assignment problem, vol. 14. (Springer, Heidelberg, 1988)
53. Encyclopedia britannica online: Surface area, volume, and average depth of oceans and seas (2008). http://www.britannica.com/eb/article-9116157/Surface-area-volume-and-average
54. Seven miles down: the story of the bathyscaph trieste (2006). http://bjsonline.com/watches/articles/0022_3.shtml
55. Russia685 (2008). http://www.globalsecurity.org/military/world/russia/685-specs.htm
56. S. Kirkpatrick, C.D.G., M.P. Vecchi, Optimization by simulated annealing. Science (New Series) **220**, 671–680 (1983)
57. J. Proakis, *Digital Communications*, 4th edn. (McGraw-Hill, New York, 2000)

Chapter 4
Spatiotemporal Coverage in Fusion-Based Sensor Networks

Rui Tan and Guoliang Xing

Abstract Wireless sensor networks (WSNs) have been increasingly available for critical applications such as security surveillance and environmental monitoring. As a fundamental performance measure of WSNs, *coverage* characterizes how well a sensing field is monitored by a network. Two facets of coverage, i.e., *spatial coverage* and *temporal coverage*, quantify the percentage of area that is well monitored by the network and the timeliness of the network in detecting targets appearing in the sensing field, respectively. Although advanced collaborative signal processing algorithms have been adopted by many existing WSNs, most previous analytical studies on spatiotemporal coverage of WSNs are conducted based on overly simplistic sensing models (e.g., the disc model) that do not capture the stochastic nature of sensing. In this chapter, we attempt to bridge this gap by exploring the fundamental limits of spatiotemporal coverage based on stochastic *data fusion* models that fuse *noisy* measurements of multiple sensors. We derive the scaling laws between spatiotemporal coverage, network density, and signal-to-noise ratio (SNR). We show that data fusion can significantly improve spatiotemporal coverage by exploiting the collaboration among sensors when several physical properties of the target signal are known. In particular, for signal path loss exponent of k (typically between 2.0 and 5.0), we prove that $\rho_f/\rho_d = \mathcal{O}(\delta^{2/k})$, where ρ_f and ρ_d are the densities of uniformly deployed sensors that achieve full spatial coverage or minimum detection delay under the fusion and disc models, respectively, and δ is SNR. Our results help

Part of this book chapter was written when Rui Tan was with Michigan State University.
The work presented in this chapter was supported in part by the National Science Foundation under grant CNS-0954039 (CAREER) and Singapore's Agency for Science, Technology and Research (A⋆STAR) under the Human Sixth Sense Programme.

R. Tan (✉)
Advanced Digital Sciences Center, Singapore, Singapore
e-mail: tanrui@adsc.com.sg

G. Xing
Michigan State University, East Lansing, MI 48824, USA
e-mail: glxing@msu.edu

H. M. Ammari (ed.), *The Art of Wireless Sensor Networks*,
Signals and Communication Technology, DOI: 10.1007/978-3-642-40066-7_4,

understand the limitations of the previous analytical results based on the disc model and provide key insights into the design of WSNs that adopt data fusion algorithms. Our analyses are verified through extensive simulations based on both synthetic data sets and data traces collected in a real deployment for vehicle detection.

1 Introduction

Recent years have witnessed the deployments of wireless sensor networks (WSNs) for many critical applications such as security surveillance [20], environmental monitoring [30], and target detection/tracking [26]. Many of these applications involve a large number of sensors distributed in a vast geographical area. As a result, the cost of deploying these networks into the physical environment is high. A key challenge is thus to predict and understand the expected sensing performance of these WSNs. A fundamental performance measure of WSNs is *coverage* that characterizes how well a sensing field is monitored by a network. The coverage of a network has two facets, i.e., *spatial coverage* and *temporal coverage*. The spatial coverage quantifies the percentage of area that is well monitored by the network. The temporal coverage quantifies the timeliness of the network in detecting targets appearing in the sensing field. Many recent studies are focused on analyzing the spatiotemporal coverage performance of large-scale WSNs [4, 23, 29, 38, 46, 50, 52].

Despite the significant progress, a key challenge faced by the research on spatiotemporal coverage is the obvious discrepancy between the advanced information processing schemes adopted by existing sensor networks and the overly simplistic sensing models widely assumed in the previous analytical studies. On the one hand, many WSN applications are designed based on *collaborative* signal processing algorithms that improve the sensing performance of a network by jointly processing the noisy measurements of multiple sensors. In practice, various stochastic *data fusion* schemes have been employed by sensor network systems for event monitoring, detection, localization, and classification [10, 13, 14, 20, 25, 26, 34, 39]. On the other hand, collaborative signal processing algorithms such as data fusion often have complex complications to the network-level sensing performance such as coverage. As a result, most analytical studies. on spatiotemporal coverage are conducted based on *overly simplistic* sensing models [3, 4, 18, 22, 23, 28, 29, 38, 46, 47, 52]. In particular, the sensing region of a sensor is often modeled as a disc with radius r centered at the position of the sensor, where r is referred to as the *sensing range*. A sensor *deterministically* detects the targets (events) within its sensing range. In Sect. 2, we will briefly survey the studies that are based on this disc model. Although such a model allows a geometric treatment to the coverage problem, it fails to capture the stochastic nature of sensing.

To illustrate the inaccuracy of the disc sensing model, we plot the sensing performance of an acoustic sensor in Fig. 1 using the data traces collected from a real vehicle detection experiment [14]. In the experiment, the sensor detects moving vehicles by comparing its signal energy measurement against a threshold (denoted by t).

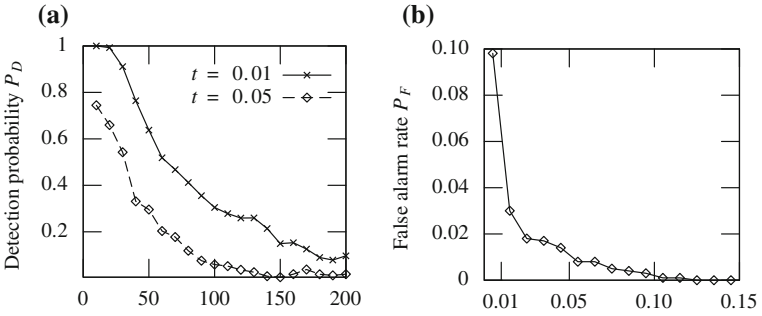


Fig. 1 Sensing performance of an acoustic sensor in detecting vehicle. **a** Detection probability versus the distance from the vehicle. **b** False alarm rate versus detection threshold

Figure 1a plots the probability that the sensor detects a vehicle (denoted by P_D) versus the distance from the vehicle. No clear cut-off boundary between successful and unsuccessful sensing of the target can be seen in Fig. 1a. Similar result is observed for the relationship between the sensor's false alarm rate (denoted by P_F) and the detection threshold shown in Fig. 1b. Note that P_F is the probability of making a positive decision when *no* vehicle is present.

In this work, we develop an analytical framework to explore the fundamental limits of spatiotemporal coverage of large-scale WSNs based on stochastic data fusion models. To characterize the inherent stochastic nature of sensing, we propose new measures for quantifying spatiotemporal coverage. Specifically, the *spatial coverage* is defined as the fraction of area in which the target can be detected with a false alarm rate of at most α and a detection probability of at least β. Similarly, to quantify the fundamental trade-off between detection delay and false alarm rate, we propose a new metric called α-*delay* that is defined as the delay of detecting a target subject to the false alarm rate bound α. The *temporal coverage* is then defined as the reciprocal of α-delay. Compared with the classical definitions of spatial and temporal coverage, these new definitions explicitly capture the performance requirements imposed by sensing applications. For instance, the full spatial coverage of a region with $\alpha = 5\,\%$ and $\beta = 90\,\%$ ensures that the probability of detecting any event occurring in the region is no lower than 90 % and no more than 5 % of the network reports are false alarms. Moreover, in the asymptotic case where α-delay is minimized, any target can be detected almost surely once after its appearance, while the false alarm rate is no greater than α.

The main focus of this paper is to investigate the fundamental scaling laws between spatiotemporal coverage, network density, and SNR. To the best of our knowledge, this work is the first to study the spatiotemporal coverage performance of large-scale WSNs based on collaborative sensing models. Our results not only help understand the limitations of the existing analytical results based on the disc model but also provide key insights into designing and analyzing the large-scale WSNs that adopt stochastic fusion algorithms. The main contributions of this work are as follows:

- We derive the spatiotemporal coverage of random networks under both data fusion and probabilistic disc models. The existing analytical results based on the classical disc model can be naturally extended to the context of stochastic event detection. With these results, we can compute the minimum network density before the deployment or turn on the fewest sensors of an existing network to achieve a desired level of spatiotemporal coverage.
- We study the fundamental scaling laws of spatiotemporal coverage. Let ρ_d and ρ_f denote the minimum network densities for achieving full spatial coverage or minimum detection delay under the disc and fusion models, respectively. For randomly deployed networks, we prove that $\rho_f = \mathcal{O}(\frac{2r^2}{R^2} \cdot \rho_d)$ where r is the radius of sensing disc and R is the fusion range within which the measurements of all sensors are fused. As fusion range can be much greater than sensing range, ρ_f is much smaller than ρ_d. This result shows that data fusion can effectively reduce the network density compared with the disc model. Furthermore, the existing analytical results based on the disc model significantly overestimate the network density required for achieving coverage.
- We study the impact of SNR on the network density when full spatial coverage or minimum detection delay is required. For randomly deployed networks, we prove that $\frac{\rho_f}{\rho_d} = \mathcal{O}(\delta^{2/k})$, where δ is SNR and k is the signal's path loss exponent that typically ranges from 2.0 to 5.0. This result suggests that data fusion is more effective in reducing the density of low-SNR network deployments, while the disc model is suitable only when the SNR is sufficiently high.
- To verify our analyses, we conduct extensive simulations based on both synthetic data sets and real data traces collected from 20 sensors. The simulation results validate our analytical results under a variety of realistic settings.

This chapter is organized as follows. Section 2 reviews the related literature on spatiotemporal coverage and detection delay. Section 3 introduces background and Sect. 4 derives the spatiotemporal coverage of WSNs. Sections 5.1 and 5.2 study the impact of data fusion on spatial and temporal coverage, respectively. Section 6 discusses the implications of results and several open issues. Section 7 presents the results of performance evaluation. Section 8 concludes this chapter.

2 Related Work

2.1 Coverage

As one of the most fundamental issues in WSNs, the coverage problem has attracted significant research attention. Previous works fall into two categories, namely theoretical analysis of coverage performance and coverage maintenance algorithms/protocols. These two categories are reviewed briefly as follows, respectively. As this chapter falls into the category of the theoretical analysis of coverage performance, our review will be mainly focused on this category.

2.1.1 Analysis of Coverage

Theoretical studies of the coverage of large-scale WSNs have been conducted in [4, 18, 22, 23, 28, 29, 38, 46, 52]. Most works [22, 23, 28, 38, 46, 52] focus on deriving the asymptotic coverage of WSNs. The k-coverage is a coverage model widely used in these studies. Specifically, a network provides k-coverage if any physical point is within the sensing range of at least k sensors.

Kumar et al. [23] consider duty-cycled WSNs that are deployed on a $\sqrt{n} \times \sqrt{n}$ grids, random uniform and Poisson with density n. Each sensor independently sleeps in each time slot with probability of p. They prove that the critical value of the function $np\pi r^2 / \log(np)$ is 1 for the event of k-coverage, where r is the sensor's sensing range. In other words, when the network size n increases, to ensure k-coverage, r has an asymptotic lower bound of $\sqrt{\frac{\log np}{np}}$.

Wan et al. [46] assume that the sensors are deployed as either a Poisson point process or a uniform point process in a square or disk region. They study two asymptotic scaling laws: (i) how the probability of k-coverage changes with the sensing range and the number of sensors, when the region to be covered is a unit square or disk; and (ii) how the probability of k-coverage changes with the area of the region to be covered and the number of sensors, when the sensors have unit sensing range. The upper and lower bounds for the probability of k-coverage are derived. Moreover, the asymptotic conditions for the k-coverage with high probability are also derived.

Shakkottai et al. [38] consider that n sensors are deployed at the grid points of a unit square area. They prove the necessary and sufficient conditions for the 1-coverage and network connectivity, i.e., $p \cdot r$ has an order of $\frac{\log n}{n}$, where p is the probability that a sensor is active. They also derive the order of the number of hop counts from any active node to another, which is $\sqrt{n / \log n}$. This study assumes that the sensing range and communication range are the same, which is a limitation of this study.

Zhang et al. [52] consider a Poisson sensor deployment with density λ in a square region with side length l, where each sensor covers a unit disk. They derive the necessary and sufficient condition of λ for k-coverage when l increases, i.e., $\lambda = \log l^2 + (k+1) \log \log l^2 + c(l)$ where $c(l) \to +\infty$ as $l \to \infty$. Based on this result, they prove that the upper bound of the network lifetime is kT where T is the lifetime of a single sensor, if $\lambda = \log l^2 + (k+1) \log \log l^2 + c(l)$ where $c(l) \to -\infty$ as $l \to \infty$.

The above studies [23, 38, 46, 52] focus on the full k-coverage over all region, i.e., every physical point is covered by at least k sensors. In [22], Kumar et al. study the k-barrier coverage problem: when an intruder crosses a belt area deployed with sensors, it can be detected with high probability by at least k sensors. Different from the full k-coverage, k-barrier coverage does not require that each physical point in the monitored region is covered by k sensors. If sensors are stealthy, the k-barrier coverage is defined as *weak* k-barrier coverage; otherwise, it is defined as *strong* k-barrier coverage. The critical conditions for *weak* and *strong* k-barrier coverage are derived by Kumar et al. in [22] and Liu et al. in [28], respectively. The critical

conditions can be used to compute the minimum number of sensors to provide barrier coverage with high probability.

Ammari et al. [2] study the critical phase transitions for coverage and connectivity based on percolation theory. Specifically, the sensing-coverage phase transition is the abrupt change from small fragmented covered areas to a single large covered area, when more sensors are continuously added to a WSN. Similarly, the network-connectivity phase transition is the abrupt change from an originally disconnected WSN to a connected WSN as more sensors are added. The covered area fractions for both transitions are derived at critical percolation.

Liu et al. [29] study the coverage performance of WSNs using other three coverage metrics, i.e., area coverage, node coverage, and detectability. The area coverage is defined as the fraction of the geographical area covered by one or more sensors. The node coverage is defined as the fraction of sensors that can be removed without reducing the area coverage. Detectability is defined as the probability that a WSN can detect an object moving along a line segment in the WSN. Liu et al. derive the closed-form formulas of these three coverage metrics for random infinite plane deployments and random strip deployments under the disc sensing model and a general sensing model that considers signal decay, respectively.

The temporal coverage, i.e., the latency of detecting a target, is another important facet of the coverage performance of WSNs. Cao et al. [5] derive the average latencies of detecting static or mobile target when sensors are deployed randomly and follow a random sleep scheduling scheme. Dousse et al. [12] address a similar problem where only the sensors with a connected path to the sink are considered. In [24], Lazos et al. map the problem of detecting mobile targets using randomly deployed sensors to a line-set intersection problem. Their analysis shows that the detection probability and the detection delay depends on the length of the perimeters of the sensing areas of sensors and not their shapes.

Most of the above theoretical results on coverage for both static and mobile sensors/targets are surveyed and compared in [4]. However, all the above theoretical studies are based on the deterministic disc model. In this chapter, we compare our results obtained under a data fusion model against the results from [4, 29].

2.1.2 Coverage Maintenance Algorithms

Early work [27, 31, 32] quantifies spatiotemporal coverage by the length of target's path where the accumulative observations of sensors are maximum or minimum [27, 31, 32]. However, these works focus on devising algorithms for finding the target's paths with certain level of coverage. Several algorithms and protocols [7, 50, 51] are designed to maintain spatiotemporal coverage using the minimum number of sensors. However, the effectiveness of these schemes largely relies on the assumption that sensors have circular sensing regions and deterministic sensing capability. Several recent studies [1, 21, 37, 48, 53] on the coverage problem have adopted probabilistic sensing models. The numerical results in [48] show that the coverage of a network can be expanded by the cooperation of sensors through data fusion. How-

ever, these studies do not quantify the improvement of coverage due to data fusion techniques. Different from our focus on analyzing the fundamental limits of coverage in WSNs, all of these studies aim to devise algorithms and protocols for coverage maintenance.

2.2 Data Fusion

There is a vast amount of literature on stochastic signal detection based on multi-sensor data fusion. Early works [6, 44] focus on small-scale powerful sensor networks (e.g., several radars). Recent studies on data fusion have considered the specific properties of WSNs such as sensors' spatial distribution [13, 14, 34] and limited sensing/communication capability [10]. However, these studies focus on analyzing the optimal fusion strategies that maximize the system performance of a given network. In contrast, this chapter explores the fundamental limits of spatiotemporal coverage of WSNs that are designed based on existing data fusion strategies. Recently, irregular sampling theory has been applied for reconstructing physical fields in WSNs [35, 36]. Different from these works that focus on developing sampling schemes to improve the quality of signal reconstruction, we aim to analyze sensors' spatial density for achieving the required level of coverage.

Many sensor network systems have incorporated various data fusion schemes to improve the system performance. In the surveillance system based on MICA2 motes [20], the system false alarm rate is reduced by fusing the detection decisions made by multiple sensors. In the DARPA SensIT project [14], advanced data fusion techniques have been employed in a number of algorithms and protocols designed for target detection [10, 26], localization [25, 39], and classification [13, 14]. Despite the wide adoption of data fusion in practice, the performance analysis of large-scale fusion-based WSNs has received little attention.

3 Preliminaries and Problem Definition

This section first introduces the preliminaries in Sect. 3.1, and then formally defines the spatiotemporal coverage of wireless sensor networks in Sect. 3.2.

3.1 Preliminaries

In this section, we describe the technical preliminaries of this chapter, which include sensor measurement, network and data fusion models.

3.1.1 Sensor Measurement Model

We assume that sensors perform detection by measuring the energy of signals emitted by the target.[1] The energy of most physical signals (e.g., acoustic and electromagnetic signals) attenuates with the distance from the signal source. Suppose sensor i is d_i meters away from the target that emits a signal of energy S_0. The attenuated signal energy s_i at the position of sensor i is given by $s_i = S_0 \cdot w(d_i)$, where $w(\cdot)$ is a decreasing function satisfying $w(0) = 1$, $w(\infty) = 0$, and $w(x) = \Theta(x^{-k})$. The $w(\cdot)$ is referred to as the *signal decay function*. Depending on the environment, e.g., atmosphere conditions, the signal's path loss exponent k typically ranges from 2.0 to 5.0 [19, 25]. We note that the theoretical results derived in this chapter do not depend on the closed-form formula of $w(\cdot)$. We adopt the following signal decay function in the simulations conducted in this chapter:

$$w(x) = \frac{1}{1 + x^k}. \tag{1}$$

The sensor measurements are contaminated by additive random noises from sensor hardware or environment. Depending on the hypothesis that the target is absent (H_0) or present (H_1), the measurement of sensor i, denoted by y_i, is given by

$$H_0 : y_i = n_i, \qquad H_1 : y_i = s_i + n_i,$$

where n_i is the energy of noise experienced by sensor i. We assume that the noise n_i at each sensor i follows the normal distribution, i.e., $n_i \sim \mathcal{N}(\mu, \sigma^2)$, where μ and σ^2 are the mean and variance of n_i, respectively. We assume that the noises, $\{n_i | \forall i\}$, are spatially independent across sensors. Therefore, the noises at sensors are independent and identically distributed (*i.i.d.*) Gaussian noises. In the presence of target, the measurement of sensor i follows the normal distribution, i.e., $y_i | H_1 \sim \mathcal{N}(s_i + \mu, \sigma^2)$. Due to the independence of noises, the sensors' measurements, $\{y_i | \forall i, H_1\}$, are spatially independent but *not* identically distributed as sensors receive different signal energies from the target. We define the PSNR as $\delta = S_0/\sigma$ which quantifies the noise level. The symbols used in this chapter are summarized in Table 1.

The above signal decay and additive *i.i.d.* Gaussian noise models have been widely adopted in the literature of multi-sensor signal detection [1, 6, 10, 25, 29, 32, 34, 39, 44, 48] and also have been empirically verified [19, 25]. In practice, the parameters of these models (i.e., S_0, $w(\cdot)$, μ, and σ^2) can be estimated using the training data collected by the existing WSN or several in situ sensors before the large-scale deployment. The normal distribution might be an approximation to the real noise distribution in practice. As discussed in Sect. 6.2, the assumption of *i.i.d.* Gaussian noises can be relaxed to any *i.i.d.* noises.

[1] Several types of sensors (e.g., acoustic sensor) only sample *signal intensity* at a given sampling rate. The *signal energy* can be obtained by preprocessing the time series of a given interval, which has been commonly adopted to avoid the transmission of raw data [10, 13, 14, 25, 39].

Table 1 Summary of notation[a]

Symbol[a]	Definition
$\mathcal{O}(\cdot)$	Asymptotic upper bound notation
$\Theta(\cdot)$	Asymptotic tight bound notation
$Q(x)$	CCDF of standard normal distribution
S_0	Original signal energy emitted by the target
μ, σ^2	Mean and variance of noise energy
δ	PSNR, $\delta = S_0/\sigma$
k	Path loss exponent
$w(\cdot)$	Signal decay function, $w(x) = \Theta(x^{-k})$
s_i	Attenuated signal energy
n_i	Noise energy, $n_i \sim \mathcal{N}(\mu, \sigma^2)$
y_i	Signal energy measurement, $y_i = s_i + n_i$
Y	Fusion statistic at cluster head / base station
P_F / P_D	False alarm rate / detection probability
α / β	Upper / lower bound of P_F / P_D
H_0 / H_1	Hypothesis that the target is absent / present
ρ	Network density
$\mathbf{F}(p)$	The set of sensors within fusion range of point p
$N(p)$	The number of sensors in $\mathbf{F}(p)$
ϵ	Upper bound of target localization error
t	Local detection threshold
T	Detection threshold at cluster head
T_D	Detection period
R	Fusion range under data fusion model
r	Disc radius under disc sensing model
c	Spatial coverage of a network
τ	Average detection delay of a network
v	Movement speed of target

[a] The symbols with subscript i refer to the notation of sensor i

3.1.2 Network Model

We consider a network deployed in a vast two-dimensional geographical region. The positions of sensors are uniformly and independently distributed in the region. Such a deployment scenario can be modeled as a stationary two-dimensional Poisson point process. Let ρ denote the density of the underlying Poisson point process. The number of sensors located in a region A, $N(A)$, follows the Poisson distribution with mean of $\rho||A||$, i.e., $N(A) \sim \text{Poi}(\rho||A||)$, where $||A||$ represents the area of the region A. We note that the uniform sensor distribution has been widely adopted in the performance analysis of large-scale WSNs [4, 23, 29, 38, 46]. Therefore, this assumption allows us to compare our results with previous analytical results.

When we analyze the temporal coverage performance of a network, we consider the following sensor sampling scheme and target mobility model. We assume that a sensor executes detection task every T_D seconds. T_D is referred to as the *detection*

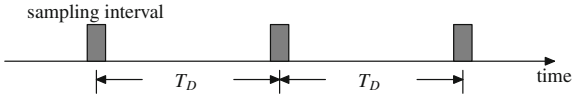


Fig. 2 Temporal view of a single sensor's operation. The sensor outputs an energy measurement after each sampling interval

period. In each detection period, a sensor gathers the signal energy during the *sampling interval* for the detection made in the current detection period. We assume that the sampling interval is much shorter than the detection period. The temporal view of a single sensor's operation is illustrated in Fig. 2. We note that such an intermittent measurement scheme is consistent with several wireless sensor systems for target detection and tracking [13, 14, 20]. For instance, a sensor may wake up every 5 seconds and sample acoustic energy for 0.05 s, where T_D is 5 s and the sampling interval is 0.05 s [14]. We assume that the target may appear at any location in the deployment region and move freely. Moreover, the target is blind to the network, i.e., the target does not know the sensors' positions, and hence it cannot choose a movement scheme to reduce the probability of being detected. The sensors synchronously detect the target, and we refer to the target detection in one detection period as the *unit detection*. The process of detecting a target consists of a series of unit detections. As the sampling interval is much shorter than the detection period, we ignore the target's movement during the sampling interval.

3.1.3 Data Fusion Model

Data fusion can improve the performance of detection systems by jointly considering the noisy measurements of multiple sensors. There exist two basic data fusion schemes, namely *decision fusion* and *value fusion*. In decision fusion, each sensor makes a *local* decision based on its measurements and sends its decision to the cluster head, which makes a *system* decision according to the local decisions. In value fusion, each sensor sends its measurements to the cluster head, which makes the detection decision based on the received measurements. In this chapter, we focus on value fusion, as it usually has better detection performance than decision fusion [44]. Most of the results in this chapter can be extended to address the decision fusion model. The details of the extensions can be found in [41, 42]. The optimal value fusion rule is to compare a weighted sum of sensors' measurements, i.e., $\sum_i \frac{s_i}{\sigma} \cdot y_i$, to a threshold [41]. However, as sensor measurements contain both noise and signal energy, the weight $\frac{s_i}{\sigma}$, i.e., the SNR received by sensor i, is unknown. A practical solution is to adopt equal constant weights for all sensors' measurements [10, 34, 48]. Since the measurements from different sensors are treated equally, the sensors far away from the target should be excluded from data fusion as their measurements suffer low SNRs. Therefore, we adopt a fusion scheme as follows.

When the network detects whether a target is present at a physical point p, the sensors within a distance of R meters from p form a cluster and fuse their measurements to detect whether a target is present at p. R is referred to as the *fusion range* and $\mathbf{F}(p)$ denotes the set of sensors within the fusion range of p. The number of sensors in $\mathbf{F}(p)$ is represented by $N(p)$. A cluster head is elected to make the detection decision by comparing the sum of measurements reported by member sensors in $\mathbf{F}(p)$ against a detection threshold T. Let Y denote the *fusion statistic*, i.e., $Y = \sum_{i \in \mathbf{F}(p)} y_i$. If $Y \geq T$, the cluster head decides H_1; otherwise, it decides H_0.

We assume that the cluster head makes a detection based on snapshot measurements from member sensors in each unit detection without using temporal samples to refine the detection decision. Such a snapshot scheme is widely adopted in previous works on target surveillance [10, 25, 34, 39, 48]. Fusion range R is an important design parameter of our data fusion model. As SNR received by sensor decays with distance from the target, fusion range lower-bounds the quality of information that is fused at the cluster head. The above data fusion model is consistent with the fusion schemes adopted in [10, 34, 48]. If more efficient fusion models are employed, the scaling laws proved in this chapter still hold as discussed in Sect. 6.2. When the network is requested to detect whether a target is present at a specified position, a cluster forms around the specified position. When the target position is not specified, we assume that the target position can be obtained through a localization algorithm. For instance, the target position can be estimated as the geometric center of a number of sensors with the largest measurements. Such a simple localization algorithm is employed in the simulations conducted in this chapter. The localized position may not be the exact target position and the distance between them is referred to as *localization error*. We assume that the localization error is upper-bounded by a constant ϵ. The localization error is accounted for in the following analyses. However, we show that it has no impact on the asymptotic results derived in this chapter. When the target is absent and the network is requested to make a detection, a cluster will still be formed and most likely yield a negative detection decision.

The above data fusion model can be used for target detection as follows. The detection can be triggered by user queries or executed periodically. In a detection process, each sensor makes a snapshot measurement and a cluster is formed by the sensors within the fusion range from the possible target to make a detection decision. The cluster formation may be initiated by the sensor that has the largest measurement. Such a scheme can be implemented by several dynamic clustering algorithms [8]. Figure 3 illustrates the intrusion detection under the data fusion model. The fusion range R can be used as an input parameter of the clustering algorithm. The communication topology of the cluster can be a multi-hop tree rooted at the cluster head. As the fusion statistic Y is an aggregation of sensors' measurements, it can be computed efficiently along the routing path to the cluster head. In this chapter, we are interested in the fundamental performance limits of spatial and temporal coverage under the fusion model and the design of clustering and data aggregation algorithms is beyond the scope of this chapter.

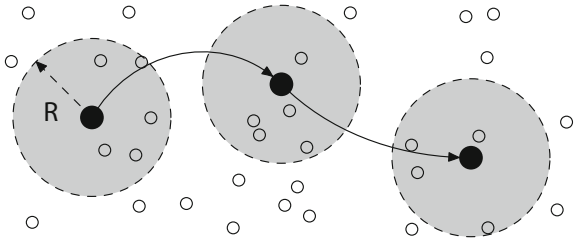


Fig. 3 Target detection under data fusion model. The *void circles* represent randomly deployed sensors; the *solid circles* represent the target in different sampling intervals, and a unit detection is performed in each sampling interval; the *dashed discs* represent the fusion ranges

3.2 Definitions and Problem Statement

3.2.1 Definition of Spatiotemporal Coverage

The detection of a target is inherently stochastic due to the noise in sensor measurements. The detection performance is usually characterized by two metrics, namely, the false alarm rate (denoted by P_F) and detection probability (denoted by P_D). P_F is the probability of making a positive decision when *no* target is present, and P_D is the probability that a present target is correctly detected. In stochastic detection, positive detection decisions may be false alarms caused by the noise in sensor measurements. In particular, although the detection probability can be improved by setting lower detection thresholds, the fidelity of detection results may be unacceptable because of high false alarm rates. Therefore, P_F together with P_D characterize the sensing quality provided by the network. For a physical point p, we denote the probability of successfully detecting a target located at p as $P_D(p)$. Note that P_F is the probability of making positive decision when *no* target is present, and hence is location independent. We first introduce a concept called (α, β)-*covered*.

Definition 1 **((α, β)-covered).** Given two constants $\alpha \in (0, 0.5)$ and $\beta \in (0.5, 1)$, a physical point p is (α, β)-*covered* if the false alarm rate P_F and detection probability $P_D(p)$ satisfy

$$P_F \leq \alpha, \quad P_{D(p)} \geq \beta.$$

We now formally define *spatial coverage* that quantifies the fraction of the surveillance region where P_F and P_D are bounded by α and β, respectively.

Definition 2 **(Spatial coverage).** The spatial coverage of a region is defined as the fraction of points in the region that are (α, β)-covered.

There also exists a fundamental trade-off between the delay of detection and false alarm rate. Although detection delay can be reduced by making sensors more sensitive (e.g., setting lower detection threshold), the fidelity of detection results may be unacceptable due to high false alarm rates. Therefore, studying detection delay

alone without the consideration of false alarm is meaningless. We now introduce a new concept called α-*delay* that quantifies the delay of detection under bounded false alarm rate.

Definition 3 (α**-delay).** α-delay is the average number of detection periods before a target is first detected subject to that the false alarm rate of the network is no greater than α, i.e., $P_F \leq \alpha$, where $\alpha \in (0, 1)$.

We now formally define *temporal coverage* that quantifies the timeliness of the network in detecting targets under bounded false alarm rate.

Definition 4 (Temporal coverage). Temporal coverage is the reciprocal of α-delay.

In addition, we define the following terminologies. The *full spatial coverage* of a region refers to the case where the spatial coverage of the region approaches one, i.e., the false alarm rate is below α and the probability of detecting a target present at *any* location is above β. The *instant detection* refers to the case where the α-delay or temporal coverage approaches one, i.e., any target can be detected almost surely in the first detection period after its appearance while the system false alarm rate is below α. In practice, mission-critical surveillance applications [14, 16, 17, 20] require that the target can be detected with a high detection probability while the network maintains a low false alarm rate. Therefore, we can set α and β accordingly to meet these requirements.

We now illustrate the spatial coverage by an example, where PSNR $\delta = 1000$ (i.e., 30 dB), $\alpha = 5\,\%$, $\beta = 95\,\%$, and $R = 50$ m. Figure 4a and b illustrate the spatial coverage under the disc and fusion models, respectively. In Fig. 4b, when a target (represented by the triangle) is present, the sensors within the fusion range from it fuse their measurements to make a detection. The gray area is (α, β)-covered, where grayscale represents the value of P_D at each point. As shown in Fig. 4a, the covered region under the disc model is simply the union of all sensing discs. As a result, when a high level of spatial coverage is required, a large number of extra sensors must be deployed to eliminate small uncovered areas surrounded by sensing discs. In contrast, data fusion can effectively expand the covered region by exploiting the collaboration among neighboring sensors.

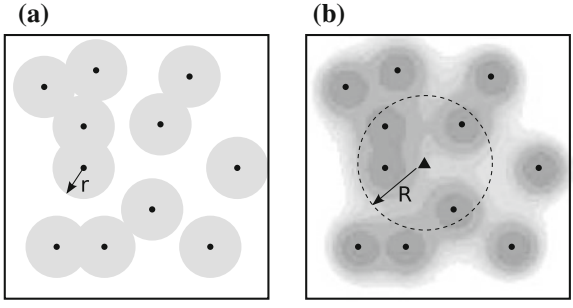


Fig. 4 Spatial coverage. **a** Spatial coverage under the disc model. Sensing range $r = 17$ m, which is computed by (4). **b** Spatial coverage under the fusion model. *Grayscale* represents the value of P_D

3.2.2 Problem Statement

In the rest of this chapter, we consider the following problems:

1. Although a number of analytical results on spatiotemporal coverage [3–5, 12, 23, 24, 29, 38, 46, 50–52] have been obtained under the classical disc model, are they still applicable under the probabilistic definition spatiotemporal coverage which explicitly captures the stochastic nature of sensing? To answer this question, we propose a probabilistic disc model such that the existing results can be naturally extended to the context of stochastic detection (Sect. 4.1).
2. How to quantify the spatiotemporal coverage when sensors can collaborate through data fusion? Answering this question enables us to evaluate the spatiotemporal coverage performance of a network. Moreover, it allows us to deploy the fewest sensors for achieving a given level of spatiotemporal coverage (Sect. 4.2).
3. What are the scaling laws between spatiotemporal coverage, network density, and SNR under both the disc and fusion models? The results will provide important insights into understanding the limitation of analytical results based on the disc model as well as the impact of data fusion on the detection performance of large-scale WSNs (Sect. 5).

4 Spatiotemporal Coverage of Wireless Sensor Networks

In this section, we derive the spatiotemporal coverage of large-scale WSNs under the disc model and the data fusion model, in Sects. 4.1 and 4.2, respectively.

4.1 Spatiotemporal Coverage under Probabilistic Disc Model

As the classical disc model deterministically treats the detection performance of sensors, existing results based on this model [3–5, 12, 23, 24, 29, 38, 46, 50–52] cannot be readily applied to analyze the performance or guide the design of real-world WSNs. In this section, we extend the classical disc model based on the stochastic detection theory [44] to capture several realistic sensing characteristics and study the spatiotemporal coverage under the extended model. The extended results will be used as the baselines to study the impact of data fusion on the sensing performance of WSNs.

4.1.1 Probabilistic Disc Model

In the *probabilistic disc model*, we choose the sensing range r such that (1) the probability of detecting any target within the sensing range is no lower than β, and (2) the false alarm rate is no greater than α. As we ignore the detection probability

outside the sensing range of a sensor, the detection capability of sensor under this model is lower than in reality. However, this model preserves the *boundary* of sensing region defined in the classical disc model. Hence, the existing results based on the classical disc model [3–5, 12, 23, 24, 29, 38, 46, 50–52] can be naturally extended to the context of stochastic detection.

We now discuss how to choose the sensing range r under the probabilistic disc model. The optimal Bayesian detection rule for a single sensor i is to compare its measurement y_i to a detection threshold t [44]. If y_i exceeds t, sensor i decides H_1; otherwise, it decides H_0. Hence, the false alarm rate P_F and detection probability P_D of sensor i are given by

$$P_F = \mathbb{P}(y_i \geq t | H_0) = Q\left(\frac{t-\mu}{\sigma}\right), \tag{2}$$

$$P_D = \mathbb{P}(y_i \geq t | H_1) = Q\left(\frac{t-\mu-s_i}{\sigma}\right), \tag{3}$$

where $\mathbb{P}(\cdot)$ is the probability notation and $Q(\cdot)$ is the complementary cumulative distribution function (CCDF) of the standard normal distribution, i.e., $Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty \mathrm{e}^{-t^2/2} \mathrm{d}\, t$. As P_D is non-decreasing function of P_F [44], it is maximized when P_F is set to be the upper bound α. Hence the optimal detection threshold can be solved from (2) as $t_{\text{opt}} = \mu + \sigma Q^{-1}(\alpha)$, where $Q^{-1}(\cdot)$ is the inverse function of $Q(\cdot)$. By replacing $t = t_{\text{opt}}$ and $s_i = S_0 \cdot w(r)$ in (3), we have

$$r = w^{-1}\left(\frac{Q^{-1}(\alpha) - Q^{-1}(\beta)}{\delta}\right), \tag{4}$$

where $w^{-1}(\cdot)$ is the inverse function of $w(\cdot)$. If the target is more than r meters from the sensor, the detection performance requirements, i.e., α and β, cannot be satisfied by setting any detection threshold. Note that a similar definition of sensing range is proposed in [48] for stochastic detection. From (4), the sensing range of a sensor varies with the user requirements (i.e., α and β) and PSNR δ. For instance, the sensing range r is 3.8 m if $\alpha = 5\,\%$, $\beta = 95\,\%$, $\delta = 50$ (i.e., 17 dB) and $w(\cdot)$ is given by (1) with $k = 2$. Note that the PSNR of 17 dB is set according to the measurements from the vehicle detection experiments based on MICA2 [16] and ExScal [17] motes. As $w(\cdot)$ is a decreasing function, $w^{-1}(\cdot)$ is also a decreasing function. Therefore, r increases with the PSNR δ according to (4). This conforms to the intuition that a sensor can detect a farther target if the noise level is lower (i.e., a greater δ).

4.1.2 Spatial Coverage Under Probabilistic Disc Model

We now extend the spatial coverage of random networks [4, 29] derived under the classical disc model to probabilistic disc model. Under both the classical and

probabilistic disc models, a location is regarded as being covered if it is within at least one sensor's sensing range. Accordingly, the area of the union of all sensors' sensing ranges is regarded as being covered by the network. The coverage of random networks under the classical disc model has been extensively studied based on the stochastic geometry theory [4, 29]. The results [4, 29] can be stated as the following lemma:

Lemma 1 *Let c denote the spatial coverage under the disc model, we have*

$$c = 1 - \mathrm{e}^{-\rho\pi r^2}, \tag{5}$$

where ρ is the network density.

If the sensing range r is chosen by (4), Eq. (5) computes the spatial coverage of a random network under the probabilistic disc model. This result will be used as the basis for studying the impact of data fusion on spatial coverage in Sect. 5.1.

4.1.3 Temporal Coverage Under Probabilistic Disc Model

Before deriving the temporal coverage under probabilistic disc model, we first introduce the target detection under the model. The network periodically detects the target as described in Sect. 3.1.2. In each unit detection, if the target is within at least one sensor's sensing range, the target is detected with a probability no lower than β. We let β be sufficiently close to 1 (e.g., $\beta = 0.99$) such that the target is detected almost surely if it is within any sensor's sensing range. Such a setting enables the sensors to exhibit similar deterministic property as under the classical disc model. We refer to the circular region with radius of r centered at the target as the *target disc*. Hence, the target is detected if there is at least one sensor within the target disc. In this section, we assume that there is no overlap between any two target discs such that the unit detections are independent from each other. Such independence among unit detections can significantly simplify the analysis. In Sect. 5.2.2, we will extend the analysis to the case where target discs may overlap. We now discuss the condition for no overlap between any two target discs. Suppose the target moves at a constant speed of v, the no-overlap condition can be satisfied if $vT_D > 2r$. For instance, if the sensing range r is 3.8 m as mentioned in Sect. 4.1.1 and the target speed v is 5 m/s (i.e., 18 km/h) [14], the target discs will no overlap as long as the detection period T_D is greater than 2 s. We have the following lemma. The proof can be found in Appendix 1.

Lemma 2 *Let τ denote the α-delay under the probabilistic disc model. If there is no overlap between any two target discs,*

$$\tau = \frac{1}{1 - \mathrm{e}^{-\rho\pi r^2}}.$$

We can see from Lemma 2 that the α-delay decreases with network density ρ and sensing range r. Note that r is given by (4) under the probabilistic disc model. With the α-delay, we can calculate the temporal coverage of the network.

4.2 Spatiotemporal Coverage Under Data Fusion Model

Although the probabilistic disc model discussed in Sect. 4.1 captures the stochastic nature of sensing, it does not exploit the collaboration among sensors. In this section, we first derive the spatiotemporal coverage of random networks under the fusion model and illustrate the analytical results using numerical examples.

4.2.1 Spatial Coverage Under Data Fusion Model

We have the following lemma regarding the spatial coverage of random networks. The proof can be found in Appendix 2.

Lemma 3 *The spatial coverage of a uniformly deployed network under the data fusion model, denoted by c, is*

$$c = \mathbb{P}\left(\frac{\sum_{i \in \mathbf{F}(p)} s_i}{\sqrt{N(p)}} \geq \sigma \left(Q^{-1}(\alpha) - Q^{-1}(\beta) \right) \right), \tag{6}$$

where p is an arbitrary physical point in the network.

As p is an arbitrary point in the network, $N(p)$ is a Poisson random variable, i.e., $N(p) \sim \text{Poi}(\rho\pi R^2)$. Moreover, $\{s_i | i \in \mathbf{F}(p)\}$ are also random variables. However, we have no closed-form formula for computing (6) due to the difficulty of deriving the cumulative distribution function (CDF) of $\frac{\sum_{i \in \mathbf{F}(p)} s_i}{\sqrt{N(p)}}$. We now give an approximation to (6) in the following lemma. The proof can be found in Appendix 3.

Lemma 4 *Let μ_s and σ_s^2 denote the mean and variance of $s_i | i \in \mathbf{F}(p)$ for arbitrary point p, respectively. The spatial coverage of a uniformly deployed network under the data fusion model can be approximated by*

$$c \simeq Q\left(\frac{\gamma(R) - \rho\pi R^2}{\sqrt{\rho\pi R^2}} \right), \tag{7}$$

where $\gamma(R) = \left(\frac{Q^{-1}(\alpha)\sigma - Q^{-1}(\beta)\sqrt{\sigma_s^2 + \sigma^2}}{\mu_s} \right)^2$.

We note that the formulas of μ_s and σ_s^2 are given by (20) and (21), respectively. As central limit theorem (CLT) is applied in the derivation of (7) (see Appendix 3), this

approximation is accurate when $N(p) \geq 20$ [33]. This condition can be easily met in many applications. For example, it is shown in [16] that the detection probability is only about 40 % when four MICA2 motes are deployed in a $10 \times 10\,\mathrm{m}^2$ region. Suppose $R = 20\,\mathrm{m}$ and the network density is the same as in [16], $N(p)$ will be about 50. With the approximate formula, we can evaluate the coverage performance of an existing network or compute the minimum network density to achieve the desired level of coverage under the fusion model. Our simulation results in Sect. 7 show that (7) can provide accurate prediction of coverage under the fusion model. We note that the localization error has little impact on the accuracy of the approximate formula when $R \gg \epsilon$. Recent sensor network localization protocols can achieve a precision within 0.5 m in large-scale outdoor deployments [43].

We now derive the lower bound of spatial coverage under the fusion model, which will be used in the derivations of scaling laws in Sect. 5.1. We denote $F_{\mathrm{Poi}}(\cdot|\lambda)$ as the CDF of the Poisson distribution $\mathrm{Poi}(\lambda)$, which is formally given by $F_{\mathrm{Poi}}(x|\lambda) = \sum_{k=0}^{\lfloor x \rfloor} \frac{\mathrm{e}^{-\lambda}\lambda^k}{k!}$. We have the following lemma. The proof can be found in Appendix 4.

Lemma 5 *The lower bound of spatial coverage of a uniformly deployed network under the data fusion model, denoted by c_L, is given by*

$$c_L = 1 - F_{\mathrm{Poi}}(\Gamma(R)|\rho\pi R^2), \tag{8}$$

where

$$\Gamma(R) = \left(\frac{Q^{-1}(\alpha) - Q^{-1}(\beta)}{\delta}\right)^2 \cdot \frac{1}{w^2(R+\epsilon)}. \tag{9}$$

When $\rho\pi R^2$ is large enough,

$$c_L = Q\left(\frac{\Gamma(R) - \rho\pi R^2}{\sqrt{\rho\pi R^2}}\right). \tag{10}$$

We now provide several numerical results to help understand the spatial coverage performance of random networks under the data fusion model. We adopt the signal decay function given by (1) with $k = 2$. Figure 5 plots the approximate coverage computed by (7). We can see from Fig. 5 that the coverage initially increases with fusion range R, but decreases to zero eventually. Intuitively, as the fusion range increases, more sensors contribute to the data fusion resulting in better sensing quality. However, as R becomes very large, the aggregate noise starts to cancel out the benefit because the target signal decreases quickly with the distance from the target. In other words, the measurements of sensors far away from the target contain low quality information and hence fusing them leads to lower detection performance. An important question is thus how to choose the optimal fusion range (denoted by R_{opt}) that maximizes the coverage. First, the R_{opt} can be obtained through numerical experiments. Figure 6 plots the optimal fusion ranges under different network densities, which are obtained by numerically maximizing the coverage. Second, it

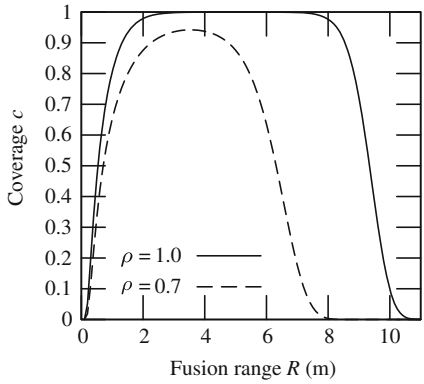


Fig. 5 Spatial coverage versus fusion range ($\delta = 4$, $\alpha = 5\,\%$, $\beta = 95\,\%$)

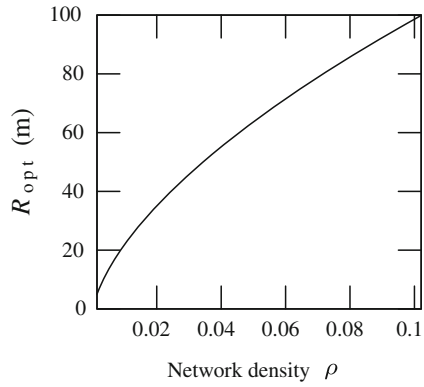


Fig. 6 Optimal fusion range versus density ($\delta = 100$, $\alpha = 5\,\%$, $\beta = 95\,\%$)

is possible to obtain the analytical R_{opt} by solving $\frac{\mathrm{d}c}{\mathrm{d}R} = 0$. For instance, when the signal decay function $w(\cdot)$ is given by (1) with $k = 2$, R_{opt} satisfies $\frac{R_{\text{opt}}}{\ln R_{\text{opt}}} = \Theta(\sqrt{\rho})$ and hence R_{opt} increases with network density ρ.

4.2.2 Temporal Coverage Under Data Fusion Model

As discussed in Sect. 3.1.2, sensors perform a unit detection in each detection period and hence the process of detecting a target consists of a series of unit detections. Denote $\mathbf{F}_j$ as the set of sensors within the fusion range in the jth unit detection. Suppose there are N_j sensors in $\mathbf{F}_j$. When no target is present, we have $Y|H_0 = \sum_{i\in\mathbf{F}_j} n_i \sim \mathcal{N}(N_j\mu, N_j\sigma^2)$, which has been proved in Lemma 3. Therefore, the false alarm rate of the jth unit detection, denoted by P_{Fj}, is given by $P_{Fj} = \mathbb{P}(Y \geq \eta|H_0) = Q\left(\frac{T - N_j\mu}{\sqrt{N_j}\sigma}\right)$, where T is the detection threshold. As P_D is a non-decreasing function of P_F [44], it is maximized when P_F is set to be the upper bound α.

Let $P_{Fj} = \alpha$, the optimal detection threshold can be derived as $T_{\text{opt}} = N_j\mu + \sqrt{N_j}\sigma Q^{-1}(\alpha)$. When the target is present, the sum of energy measurements in the jth unit detection approximately follows a normal distribution $Y|H_1 = \sum_{i\in\mathbf{F}_j} s_i + \sum_{i\in\mathbf{F}_j} n_i \sim \mathcal{N}(N_j\mu_s + N_j\mu, N_j\sigma_s^2 + N_j\sigma^2)$, which has been proved in Lemma 4. The detection probability in the jth unit detection, denoted by P_{Dj}, is given by $P_{Dj} = \mathbb{P}(Y \geq T|H_1) \simeq Q\left(\frac{T - N_j\mu_s - N_j\mu}{\sqrt{N_j}\cdot\sqrt{\sigma_s^2+\sigma^2}}\right)$. By replacing T with the optimal detection threshold T_{opt}, we have

$$P_{Dj} \simeq Q\left(\frac{\sigma}{\sqrt{\sigma_s^2+\sigma^2}}\cdot Q^{-1}(\alpha) - \frac{\mu_s}{\sqrt{\sigma_s^2+\sigma^2}}\cdot\sqrt{N_j}\right). \tag{11}$$

Based on the above performance modeling of each unit detection, we now derive the α-delay under the data fusion model. In this section, we assume that there is no overlap between any two fusion ranges (as shown in Fig. 3). As a result, the sensor sets $\{\mathbf{F}_j|j \geq 1\}$ are independent from each other. Such independence can significantly simplify the analysis. In Sect. 5.2.2, we extend the analysis to the case where fusion ranges may overlap. We now discuss the condition for no overlap between any two fusion ranges. Suppose the target moves at a constant speed of v, the no-overlap condition can be satisfied if $vT_D > 2R$. For instance, if the fusion range R is set to be 10 m and the target speed v is 5 m/s (i.e., 18 km/h) [14], the fusion ranges will not overlap as long as the detection period T_D is greater than 4 s.

From (11), P_{Dj} is a function of N_j. When the sensor sets $\{\mathbf{F}_j|j \geq 1\}$ are independent, $\{P_{Dj}|j \geq 1\}$ are *i.i.d.* as the numbers of sensors involved in each unit detection (i.e., $\{N_j|j \geq 1\}$) are *i.i.d.* due to the Poisson process. We denote $\mathbb{E}[P_D]$ as the mean of P_{Dj} for any j, i.e., $\mathbb{E}[P_D] = \mathbb{E}[P_{Dj}], \forall j$. Intuitively, the intrusion detection can be viewed as a series of infinite Bernoulli trials and the success probability of each Bernoulli trial is $\mathbb{E}[P_D]$. Accordingly, the number of unit detections before (and including) the first successful unit detection follows the geometric distribution with a mean of $1/\mathbb{E}[P_D]$. Hence the α-delay is given by the following lemma. The proof can be found in Appendix 5.

Lemma 6 *Let τ denote the α-delay of fusion-based detection. If there is no overlap between any two fusion ranges, $\tau = 1/\mathbb{E}[P_D]$, where $\mathbb{E}[P_D]$ is the average detection probability in any unit detection.*

We now discuss how to compute $\mathbb{E}[P_D]$ in Lemma 6. As P_{Dj} is a function of N_j and N_j follows the Poisson distribution, i.e., $N_j \sim \text{Poi}(\rho\pi R^2)$, $\mathbb{E}[P_D]$ is given by

$$\mathbb{E}[P_D] = \sum_{N_j=0}^{\infty} P_{Dj}\cdot f_{\text{Poi}}(N_j|\rho\pi R^2), \tag{12}$$

where $f_{\text{Poi}}(k|\lambda)$ is the probability density function (PDF) of the Poisson distribution $\text{Poi}(\lambda)$. Specifically, $f_{\text{Poi}}(k|\lambda) = \lambda^k e^{-\lambda}/k!$. Note that P_{Dj} in (12) is computed

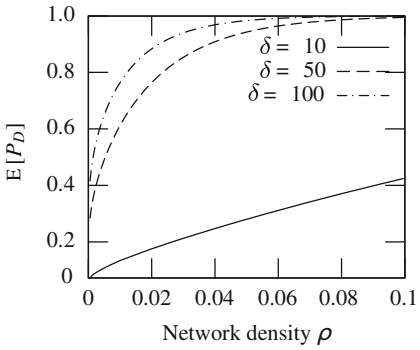


Fig. 7 Mean detection probability versus network density ($R = 25$ m)

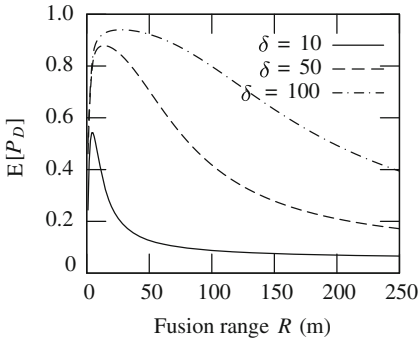


Fig. 8 Mean detection probability versus fusion range ($\rho = 0.03$)

using (11). Figures 7 and 8 plot $\mathbb{E}[P_D]$ versus network density ρ and fusion range R, respectively. From Fig. 7, we can see that $\mathbb{E}[P_D]$ increases with ρ. Moreover, for a certain ρ, $\mathbb{E}[P_D]$ increases with the PSNR. From Fig. 8, we can see that $\mathbb{E}[P_D]$ is a concave function of fusion range R and there exists an optimal R that maximizes $\mathbb{E}[P_D]$. When the fusion range initially increases, more sensors contribute to the data fusion resulting in better sensing quality. However, when the fusion range becomes very large, the aggregate noise starts to cancel out the benefit because the target signal decreases rapidly with the distance from the target. In other words, the measurements of sensors far away from the target contain low-quality information and hence fusing them lowers detection performance. In practice, we can choose the optimal fusion range according to numerical results.

5 Impact of Data Fusion on Spatiotemporal Coverage

In this section, we study the impact of data fusion on spatial coverage and temporal coverage in Sects. 5.1 and 5.2, respectively.

5.1 Impact of Data Fusion on Spatial Coverage

Many mission-critical applications require a high level of spatial coverage over the surveillance region. As an asymptotic case, full spatial coverage is required, i.e., *any* target/event present in the region can be detected with a probability of at least β while the false alarm rate is below α. For random networks, a higher level of coverage always requires more sensors. Therefore, the network density for achieving full spatial coverage is an important cost metric for mission-critical applications.

Under the disc model, the sensing regions of randomly deployed sensors inevitably overlap with each other when a high level coverage is required. According to (5), we have $\mathrm{d}\rho = \frac{1}{\pi r^2} \cdot \frac{1}{1-c} \cdot \mathrm{d}c$. If c is close to 1, a large number of extra sensors (i.e., $\mathrm{d}\rho$) are required to eliminate a small uncovered area (i.e., $\mathrm{d}c$). Moreover, the situation gets worse when c increases. In this section, we are interested in how much network density can be reduced by adopting data fusion. Specifically, we study the asymptotic relationships between the network densities for achieving full spatial coverage under the probabilistic disc and data fusion models. The results provide important insights into understanding the limitation of the disc model and the impact of data fusion on spatial coverage of random networks.

5.1.1 Full Spatial Coverage Using Fixed Fusion Range

We first study the relationship between the network densities for achieving full spatial coverage under the disc and fusion models when fusion range R is a constant. We have the following theorem. The proof can be found in Appendix 6.

Theorem 1 *For uniformly deployed networks, let ρ_d and ρ_f denote the minimum network densities required to achieve the spatial coverage of c under the disc and fusion models, respectively. If the fusion range R is fixed, we have*

$$\rho_f = \mathcal{O}\left(\frac{2r^2}{R^2} \cdot \rho_d\right), \quad c \to 1^-. \tag{13}$$

Theorem 1 shows that in order to achieve full spatial coverage, ρ_f is smaller than ρ_d if $R > \sqrt{2}r$. According to (4), sensing range r is a constant independent of network density. On the other hand, fusion range R is a design parameter of the fusion model, which is mainly constrained by the communication overhead. In practice, the condition $R > \sqrt{2}r$ can be easily satisfied. For instance, the acoustic sensor on MICA2 motes has a sensing range of 3–5 m if a high performance (e.g., $\alpha = 5$ and $\beta = 95\,\%$) is required [16]. On the other hand, the fusion range can be set to be much larger. For example, Fig. 6 shows that R_{opt} ranges from 5 to 100 m when network density increases from 1.5×10^{-3} to 0.1. Therefore, according to Theorem 1, the fusion model with the optimal fusion range can significantly reduce network density for achieving a high level of coverage.

5.1.2 Full Spatial Coverage Using Optimal Fusion Range

As discussed in Sect. 4.2.1, we can obtain the optimal fusion range via numerical experiment or analysis. Data fusion with the optimal fusion range allows the maximum number of informative sensors to contribute to the detection. The scaling law obtained with optimal fusion range will help us understand the maximum performance gain by adopting the data fusion model. The following theorem shows that ρ_f further reduces to $\mathcal{O}(\rho_d^{1-1/k})$ as long as the fusion range is optimal. The proof can be found in Appendix 7.

Theorem 2 *For uniformly deployed networks, let ρ_d and ρ_f denote the minimum network densities required to achieve the spatial coverage of c under the disc and fusion models, respectively. If the optimal fusion range R_{opt} is adopted, we have*

$$\rho_f = \mathcal{O}\left(\rho_d^{1-1/k}\right), \quad c \to 1^-. \tag{14}$$

Theorem 2 shows that if the optimal fusion range is adopted, the fusion model can significantly reduce the network density for achieving high coverage. In particular, from Theorem 2, the density ratio $\frac{\rho_f}{\rho_d} = \mathcal{O}(\rho_d^{-1/k}) = 0$ when $c \to 1^-$, which means ρ_f is insignificant compared with ρ_d for achieving high coverage. Theorem 2 is applicable to the scenarios where the physical signal follows the power law decay with path loss exponent k, which are widely assumed and verified in practice. We note that the path loss exponent k typically ranges from 2.0 to 5.0 [19, 25]. In particular, the propagation of acoustic signals in free space follows the inverse-square law, i.e., $k = 2$, and therefore $\rho_f = \mathcal{O}(\sqrt{\rho_d})$.

5.1.3 Impact of Signal-to-Noise Ratio

In this section, we study the impact of PSNR on the results derived in the previous sections. PSNR is an important system parameter which is determined by the property of target, noise level, and sensitivity of sensors. We have the following theorem.

Theorem 3 *For uniformly deployed networks, if the fusion range R is fixed, we have*

$$\frac{\rho_f}{\rho_d} = \mathcal{O}(\delta^{2/k}), \quad c \to 1^-. \tag{15}$$

Proof As $w(x) = \Theta(x^{-k})$, $w^{-1}(x) = \Theta(x^{-1/k})$. According to (4), the sensing range $r = \Theta(\delta^{1/k})$. As $\lim_{c\to 1^-} \frac{\rho_f}{\rho_d} \le \frac{2r^2}{R^2} = \Theta(\delta^{2/k})$, we have (15). □

Theorem 3 suggests that for a fixed R, the relative cost between the fusion and disc models is affected by the PSNR δ. Specifically, the fusion model requires fewer sensors to achieve full spatial coverage than the disc model if the PSNR is low. On the other hand, the disc model suffices only if the PSNR is *sufficiently* high.

Intuitively, sensor collaboration is more advantageous when the PSNR is low to moderate. However, when the PSNR is *sufficiently* high, the detection performance of a single sensor is satisfactory and the collaboration among multiple sensors may be unnecessary.

5.2 Impact of Data Fusion on Temporal Coverage

Many mission-critical real-time applications require detection delay to be as small as possible [20, 45]. As an asymptotic case, the α-delay approaches one, i.e., any target can be detected almost surely in the first detection period after its appearance, which is referred to as the *instant detection*. As a smaller detection delay always requires more sensors, the network density for achieving instant detection is an important cost metric for mission-critical real-time sensor networks. In this section, we investigate the required network density for achieving instant detection under both the disc and fusion models. According to Lemma 2 and 6, the network density under both models approaches infinity[2] when the α-delay reduces to one. However, the speed that the network density increases is different. In this section, we study the ratio of network densities for instant detection under the two models, which characterize the relative cost of the two models when detection delay is minimized. The result provides important insights into understanding the limitation of the disc model and the impact of data fusion on the performance of real-time WSNs for intrusion detection. In the rest of this section, we first discuss the case that the target discs and fusion ranges under the disc and fusion models do not overlap in Sect. 5.2.1, and then generalize the results in Sect. 5.2.2.

5.2.1 Network Density for Achieving Instant Detection

We have the following lemma. The proof can be found in Appendix 8.

Lemma 7 *Let ρ_f and ρ_d denote the network densities for achieving α-delay of τ under the fusion and disc models, respectively. If there is no overlap between target discs and fusion ranges under the two models, respectively, there exists $\xi \in (0, 1)$ such that*

$$\frac{2}{\gamma^2 R^2} \cdot r^2 \leq \lim_{\tau \to 1^+} \frac{\rho_f}{\rho_d} \leq \frac{2}{\xi \gamma^2 R^2} \cdot r^2, \tag{16}$$

where $\gamma = -\frac{\mu_s}{\sqrt{\sigma_s^2 + \sigma^2}}$.

[2] Numerically, the network density ρ will not be very large when the α-delay approaches one. For instance, according to Lemma 2, suppose the sensing range r is 5 m, the α-delay under the disc model is $1 + 10^{-5}$ when $\rho = 0.15$.

Note that ξ is a function of γ (given by (30)). According to Lemma 7, $\lim_{\tau\to 1^+} \rho_f / \rho_d$ is largely affected by the sensing range of a single sensor. According to (4), the sensing range r is determined by the requirements on false alarm rate and detection probability (i.e., α and β), as well as the PSNR δ. Moreover, as discussed in Sect. 4.1.3, β is a constant close to one. Accordingly, we only analyze the impacts of α and δ on the network density for achieving instant detection. We have the following theorem. The proof can be found in Appendix 9.

Theorem 4 *If there is no overlap between target discs and fusion ranges under the disc and fusion models, respectively, for given path loss exponent k, the ratio of network densities for instant detection under the two models has an asymptotic tight bound of*

$$\frac{\rho_f}{\rho_d} = \Theta\left(\left(\frac{\delta}{Q^{-1}(\alpha)}\right)^{2/k}\right), \quad \tau \to 1^+. \tag{17}$$

Theorem 4 suggests that, for a certain path loss exponent k, the relative cost for instant detection between the fusion and disc models depends on the required false alarm rate α and PSNR δ. First, when $\alpha \to 0$, $Q^{-1}(\alpha) \to \infty$ and hence $\lim_{\tau\to 1^+} \rho_f/\rho_d \to 0$. It suggests that data fusion can significantly reduce network density when a small false alarm rate is required. Second, $\lim_{\tau\to 1^+} \rho_f/\rho_d$ increases with δ, which suggests that the advantage of data fusion diminishes as the PSNR increases. Moreover, the path loss exponent k determines the order of density ratio with regard to the PSNR. Intuitively, sensor collaboration is more advantageous when the PSNR is low. However, when the PSNR is *sufficiently* high, the detection performance of a single sensor is satisfactory and the collaboration among multiple sensors may be unnecessary.

5.2.2 Extension to General Target Speed and Detection Period

In previous sections, we assume that there is no overlap between any two target discs and fusion ranges under the disc and fusion models, respectively. However, fusion ranges may overlap if the target speed is low or the detection period T_D is short, as illustrated in Fig. 9. In this section, we will generalize the previous analyses without the no-overlap limitation. When there is no overlap, the unit detections are independent from each other. As a result, the index of first successful unit detection (i.e., J) follows the geometric distribution and the α-delay can be computed as the mean of the geometric distribution. In contrast, when target discs or fusion ranges can overlap, the detection results in different unit detections are statistically *correlated* due to the possible common sensors shared by different unit detections. Hence, J does not follows the geometric distribution anymore. Therefore, the correlation among unit detections substantially complicates the analysis of α-delay. As a result, it is difficult to obtain the closed-form formula of α-delay. Instead, we aim to find the bound of α-delay in this section. The lower bound of α-delay under the disc model is given by the following lemma. The proof can be found in Appendix 10.

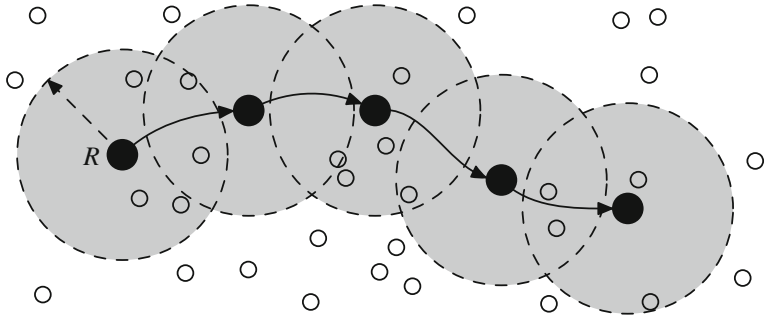


Fig. 9 The overlap case under the data fusion model. The *void circles* represent sensors; the *solid circles* represent the target in different sampling intervals; the *dashed discs* represent the fusion ranges

Lemma 8 *Let τ denote the α-delay under the probabilistic disc model. We have*

$$\tau \geq \frac{1}{1 - \mathrm{e}^{-\rho\pi r^2}}.$$

Compared with the results in Lemmas 2 and 8, we can see that the α-delay is minimized for the no-overlap case. Intuitively, the area covered by the union of target discs is maximized in the no-overlap case, which yields the maximum overall detection probability for a given number of detection periods and in turn leads to the minimum detection delay.

The upper bound of α-delay under the data fusion model is given by the following lemma. The proof can be found in Appendix 11.

Lemma 9 *Let τ denote the α-delay of fusion-based detection. We have $\tau \leq \mathbb{E}[1/P_D]$, where P_D is the detection probability in any unit detection.*

As $1/P_D$ is a convex function of P_D, according to Jensen's inequality, $\mathbb{E}[1/P_D] \geq 1/\mathbb{E}[P_D]$, where $1/\mathbb{E}[P_D]$ is the α-delay when there is no overlap between any two fusion ranges. We now discuss how to compute $\mathbb{E}[1/P_D]$ in Lemma 9. As P_{Dj} is a function of N_j which follows the Poisson distribution, i.e., $N_j \sim \mathrm{Poi}(\rho\pi R^2)$, $\mathbb{E}[1/P_D]$ can be numerically computed by averaging $\frac{1}{P_{Dj}}$ over the distribution of N_j.

With the lower and upper bounds of α-delay under the disc and fusion models, respectively, we can derive the asymptotic bound of ratio of network densities required by the two models to achieve instant detection. As it is more challenging to handle the expression $\mathbb{E}[1/P_D]$ in Lemma 9 than $\mathbb{E}[P_D]$ in Lemma 6, we will employ substantially different technique to analyze the density ratio. We have the following theorem. The proof can be found in Appendix 12.

Theorem 5 *Let ρ_f and ρ_d denote the network densities for achieving α-delay of τ under the value fusion and disc models, respectively. For given path loss exponent k, the ratio of network densities for instant detection has an asymptotic upper bound of*

$$\lim_{\tau \to 1^+} \frac{\rho_f}{\rho_d} = \mathcal{O}\left(\left(\frac{\delta}{Q^{-1}(\alpha)}\right)^{2/k}\right). \tag{18}$$

Different from the result in Theorem 4 which is the asymptotic *tight* bound of the density ratio, Theorem 5 gives the asymptotic *upper* bound. In Sect. 7.2.3, we will compare the density ratios under the overlap and no-overlap cases through simulations. Moreover, as target speed is an important factor of the overlap/no-overlap condition, we also evaluate the impact of target speed on the density ratio.

6 Implications of Results and Discussions

In this section, we first summarize the implications of the theoretical results derived in previous sections, which provide important insights into understanding the applicability of the disc model and the data fusion model in various application scenarios. We then discuss several issues that have not been addressed.

6.1 Implications of Results

6.1.1 Data Fusion Reduces Network Density

According to Theorem 2, when the coverage of random networks approaches one, ρ_d increases significantly faster than ρ_f, especially for a small path loss exponent. For instance, when $k = 2$ (which typically holds for acoustic signals), $\rho_f = \mathcal{O}(\sqrt{\rho_d})$. This result implies that the existing analytical results based on the disc model (e.g., [4, 23, 29, 46, 52]) significantly overestimate the network density required for achieving full spatial coverage of random networks. Data fusion can reduce network density for achieving instant detection as well. According to Theorem 4, when the detection delay is minimized (i.e., $\tau \to 1^+$), $\rho_f/\rho_d \to 0$ when $\alpha \to 0$. Therefore, if a small α is required, $\rho_f < \rho_d$ for instant detection, i.e., the fusion model requires lower network density than the disc model. In other words, data fusion is effective in reducing detection delay and false alarms. For instance, Fig. 10 plots the lower and upper bounds of the density ratio when the α-delay is minimized, which is given by Lemma 7. We set the PSNR δ to be 50 (i.e., 17 dB) according to the measurements in the vehicle detection experiments based on MICA2 [16] and ExScal [17] motes. The fusion range R is optimized to be 37 m. From the figure, we can see that if $\alpha < 0.2$, the fusion model outperforms the disc model. In practice, most mission-critical surveillance systems require a small α. For example, in the vehicle detection system [20] and the acoustic shooter localization system [45], the false alarm rates are tuned to be near zero. Therefore, data fusion can significantly reduces the network density for these mission-critical surveillance systems.

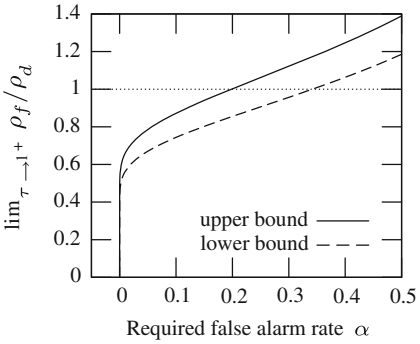


Fig. 10 Density ratio versus required false alarm rate ($k = 2$, $\delta = 50$, $R = 37$ m)

6.1.2 Disc Model Suffices for High-SNR Detection

On the other hand, Theorem 3 shows that the disc model may lead to similar or even lower network density than the fusion model for achieving full spatial coverage if PSNR is sufficiently high. Similarly, according to Theorem 4, $\lim_{\tau \to 1^+} \rho_f / \rho_d$ increases with δ for fixed α. Therefore, if the PSNR is high enough such that $\lim_{\tau \to 1^+} \rho_f / \rho_d > 1$, the disc model is superior to the fusion model in achieving instant detection. For instance, Fig. 11 plots the upper bound of density ratio versus SNR under various path loss exponents when α-delay is minimized, which is given by Lemma 7. From the figure, we can see linear and concave relationships between the density ratio and PSNR when k is 2.0 and 4.0, respectively, which are consistent with Lemma 6. Moreover, if the PSNR is sufficiently high (e.g., 22 dB), the disc model outperforms the fusion model. However, the noise experienced by a sensor comes from various sources, e.g., the random disturbances in the environ-

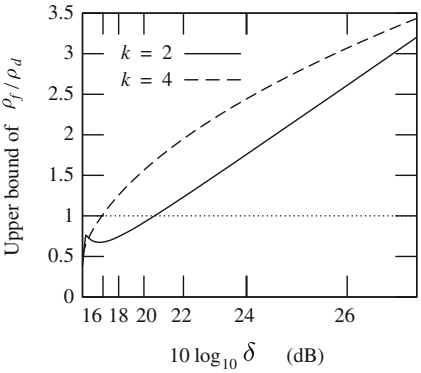


Fig. 11 Upper bound of density ratio versus SNR ($\alpha = 0.1\,\%$)

ment and the electronic noise in the sensor's circuit. Thus, the PSNR depends on the characteristics of targets, the environment, and the sensor device. In the vehicle detection experiments based on low-power motes, e.g., MICA2 [16] and ExScal [17], the PSNRs are usually low to moderate ($\leq$17 dB). In such a case, data fusion can effectively reduce the network density required to achieve a high level of coverage or a short detection delay.

6.1.3 Design of Data Fusion Algorithms

Our results provide several important guidelines on the design of data fusion algorithms for large-scale WSNs. First, data fusion is very effective in reducing network density for achieving a high level of coverage or a short detection delay. In particular, Theorems 3–5 suggest that the performance gain of data fusion increases when the PSNR is lower. Therefore, data fusion should be employed for low-SNR deployments when a high level of coverage or a short detection delay is required. Second, Theorems 1, 2, and Lemma 7 suggest that fusion range plays an important role in the achievable performance of data fusion. Particularly, as discussed in Sect. 4.2.1, the optimal fusion range that maximizes the spatial coverage of random networks increases with network density and can be numerically computed. However, a larger fusion range may lead to longer transmission distances and more sensors that take part in data fusion. Investigating the optimal fusion range under joint constraints of coverage, detection delay and communication is left for the future work.

6.2 Discussions

We now discuss several issues that have not been addressed.

6.2.1 Noise Models

In the proofs of Lemma 3, 5 and 5 the fusion statistic Y has a component $\sum_{i\in\mathbf{F}(p)} n_i$. According to the CLT, this component approximately follows the normal distribution if $\{n_i\}$ are *i.i.d.*. Therefore, the assumption of *i.i.d.* Gaussian noises made in Section 3.1.1 can be relaxed to *i.i.d.* noises that follow any distribution, when the number of sensors taking part in data fusion is large enough. In practice, the accuracy of this approximation is satisfactory when $N(p) \geq 20$ [33]. In particular, the distribution of noise will not affect the asymptotic scaling laws in Sects. 5.1 and 5.2, as $N(p)$ is large in the asymptotic scenarios where $c \to 1^-$.

6.2.2 Signal Decay Laws

The main objective of this chapter is to explore the fundamental limits of coverage and detection delay based on data fusion model in target surveillance applications, in which sensors measure the signals emitted by the target. The proofs of all lemmas and Theorem 1 are not dependent on the form of the signal decay function $w(\cdot)$. Therefore, these results hold under *arbitrary* bounded decreasing function $w(\cdot)$. However, Theorems 2–5 are only applicable for the applications where the target signal follows the power law decay, i.e., $w(x) = \Theta(x^{-k})$. We acknowledge that most mechanical and electromagnetic waves follow the power law decay in propagation. In particular, in open space, inverse-square law (i.e., $k = 2$) [11] applies to various physical signals such as sound, light, and radiation. We note that if a sensor is lifted above the ground, its received signal energy can be affected by the height. However, as Theorems 2–5 only depend the asymptotic power law decay, they still hold if the height only introduces constant gain coefficient to the decay model. In the future work, we will investigate if the height can lead to an asymptotic decay model that is different from the power law decay. Moreover, we will extend our analyses to address other decay laws such as exponential decay in diffusion processes [40].

6.2.3 Data Fusion Models

Theorems 1, 2, 3, and 5 give the upper bounds of network density under the fusion model presented in Sect. 3.1.3. If more efficient fusion models are employed, the coverage performance as well as detection delay will be further improved. Therefore, more efficient fusion model can reduce the network density for achieving a certain level of coverage or detection delay. As a result, the upper bounds of network density derived in this chapter still hold. Exploring the impact of efficiency of fusion models on network density is left for future work.

7 Evaluation

In this section, we conduct extensive simulations based on real data traces as well as synthetic data to evaluate the spatiotemporal coverage in non-asymptotic and asymptotic cases, respectively.

7.1 Trace-Driven Simulations

7.1.1 Methodology and Settings

We first conduct simulations using the data traces collected in in the DARPA SensIT vehicle detection experiment [14]. In the experiments, 75 WINS NG 2.0 nodes are deployed to detect amphibious assault vehicles (AAVs) driving through the surveillance region. We refer to [14] for detailed setup of the experiments. The dataset used in our simulations includes the ground truth data and the acoustic time series recorded by 20 nodes at a frequency of 4960 Hz when a vehicle drives through. The ground truth data include the positions of sensors and the trajectory of the AAV recorded by a global positioning system (GPS) device.

Sensors' sensing ranges under the probabilistic disc model are determined individually to meet the detection performance requirements ($\alpha = 5\%$, $\beta = 95\%$). The resulted sensing ranges are from 22.5 to 59.2 m with the average of 43.2 m. Such a significant variation is due to several issues including poor calibration and complex terrain. In our simulation, we deploy random or regular networks with size of $1000 \times 1000\,\text{m}^2$. Each sensor in the simulation is associated with a real sensor chosen at random. For each deployment, we evaluate the spatial coverage and α-delay under both the disc and fusion models, respectively.

For evaluating spatial coverage, we divide the region into 1000×1000 grids. Under the disc model, the coverage is estimated as the ratio of grid points that are covered by discs. Under the fusion model, the coverage is estimated as the ratio of (α, β)-covered grid points. Specifically, for a target that appears at a grid point, each sensor makes a measurement which is set to be the signal energy gathered by the associated real sensor at a similar distance to vehicle in the data trace. A cluster is formed around the sensor with the highest reading, which fuses sensor measurements for detection.

For evaluating α-delay, the target initially appears at the origin, and moves along the X-axis at a speed of 10 m/s. The detection period T_D is set to be 60 s. Under the disc model, once the target enters the sensing range of a sensor, the sensor makes a detection decision by comparing its measurement against the detection threshold t derived in Sect. 4.1.1. Under the fusion model, sensors fuse their measurements to detect the target as discussed in Sect. 3.1.3. The α-delay is computed as the average number of detection periods before the target is first detected in each run.

7.1.2 Simulation Results

Figure 12 plots the the numbers of uniformly deployed sensors under the disc and fusion models as well as the corresponding density ratio versus the achieved spatial coverage. We can see that the disc model suffices if a moderate level of coverage is required. However, the fusion model is more effective for achieving high coverage. In particular, the fusion model with a fusion range of 200 m saves more than 50%

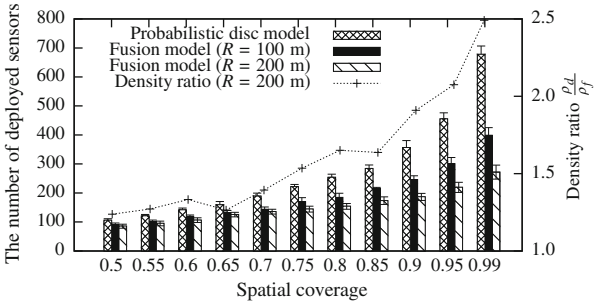


Fig. 12 The number of deployed sensors in random networks versus achieved spatial coverage

sensors when the coverage is greater than 0.75. We note that the average number of sensors taking part in data fusion is within 30 and hence will not introduce high communication overhead. According to Theorem 1, the limit of $\frac{\rho_d}{\rho_f}$ is $\frac{R^2}{2r^2}$ when the coverage approaches one. We will evaluate the coverage performance in asymptotic case through simulations based on synthetic data in Sect. 7.2. Figure 13 plots the network density versus the achieved α-delay under various settings. We can see that the fusion model is more effective than the disc model for achieving short α-delay. In particular, the fusion model with a fusion range of 100 m saves more than 50 % sensors when the α-delay is less than 2. We note that the average number of sensors taking part in data fusion is within 20 and hence will not introduce high communication overhead.

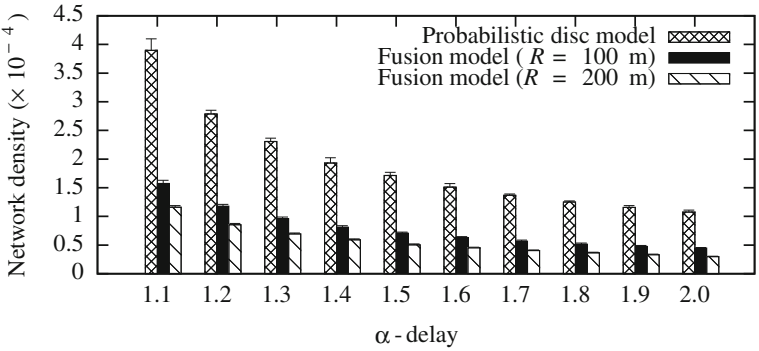


Fig. 13 The network density versus achieved α-delay

7.2 Simulations Based on Synthetic Data

7.2.1 Numerical Settings

In addition to trace-driven simulations, we also conduct extensive simulations based on synthetic data. These simulations allow us to evaluate the theoretical results in a wide range of settings. We adopt the signal decay function in (1) with $k = 2$. Both the mean and variance of the Gaussian noise generator, μ and σ^2, are set to be 1. We set the target's source energy, i.e., S_0, to be 4, 50, and 5, 000, so that the SNRs in the simulations are consistent with several real experiments [9, 14, 16, 17].

For evaluating coverage, as proved in Lemma 3, it suffices to measure the probability that a point is covered for evaluating the coverage of a random network. Hence, we let the target appear at a fixed point p and deploy random networks with size of $4R \times 4R$ centered at p. For each deployment, $P_D(p)$ is estimated as the fraction of succesful detections. The spatial coverage is estimated as the fraction of deployments whose $P_D(p)$ is greater than β. We also evaluate the impact of localization error by integrating a simple localization algorithm. Specifically, for each detection, if a sensor's reading exceeds $S_0 \cdot w(R) + \mu$, it will take part in the target localization. The target is localized as the geometric center of the sensors participating in the localization. For a regular network, it suffices to measure the fraction of covered area in a grid for evaluating the coverage of the whole network. In our simulations, we find the minimum network density with which 10×10 points in the grid are covered.

For evaluating detection delay, the target initially appears at the origin, and moves along the X-axis at a speed of $2R$ per detection period. We evaluate the impact of constant target localization error as follows. Suppose the real target position is at $P(x, y)$ when sensors take measurements, while the target position localized by the network is at $P'(x + \epsilon \cos\theta, y + \epsilon \sin\theta)$, where ϵ is a specified constant and θ is picked uniformly from $[0, 2\pi)$. Sensors within the fusion range centered at P' fuse their measurements and make the detection decision. We also evaluate the impact of the overlap/no-overlap condition by comparing the simulation results under the overlap and no-overlap cases. For the overlap case, the target moves $\frac{R}{2}$ and $\frac{r}{2}$ in each detection period under the fusion and disc models, respectively; for the no-overlap case, it moves $2R$ and $2r$, respectively.

7.2.2 Spatial Coverage

We first present the simulation results if sensors are randomly deployed. The first set of simulations evaluate the accuracy of the approximate formula given in Lemma 4. Figure 14 plots the analytical and measured coverage versus network density. The curves labelled with SIM-LOC and SIM represent the measured results with and without accounting for localization error, respectively. We can see that the simulation result matches well the analytical result given by (7). A network density of 0.8 is enough to provide high coverage under the fusion model, where the SNR is very

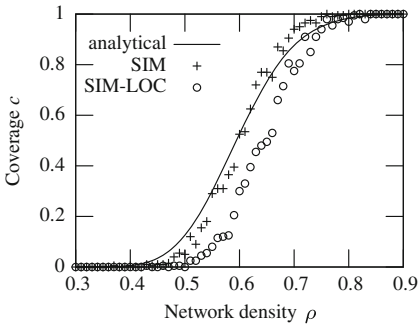


Fig. 14 Spatial coverage of random networks versus network density ($\delta = 4$, $R = 5\,\text{m}$)

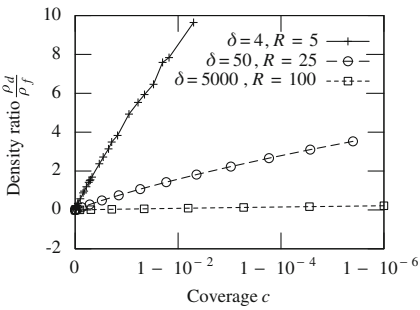


Fig. 15 Density ratio $\frac{\rho_d}{\rho_f}$ of random networks versus spatial coverage in $\log_{10}$ scale with various PSNRs

low ($\delta = 4$). When there is localization error, a maximum deviation of about 0.2 from the analytical result can be seen from Fig. 14. The coverage decreases in the presence of localization error as sensors received weaker signals when the target cannot be accurately localized. However, the impact of localization error diminishes when $c \to 1^-$.

The second set of simulations evaluate the impact of SNR on the asymptotic network densities. Figure 15 plots the network density ratio $\frac{\rho_d}{\rho_f}$ versus the achieved coverage under various PSNRs, where ρ_d is computed by (5) and ρ_f is obtained in simulations, respectively. The x-axis is plotted in $\log_{10}$ scale. We can see that the density ratio increases with the coverage, i.e., the fusion model becomes more effective for achieving higher coverage. Moreover, the density ratio decreases with the PSNR, which conforms to the result of Theorem 3. For instance, to achieve a high coverage of 0.99, the density ratio $\frac{\rho_d}{\rho_f}$ is about 8 when $\delta = 4$. The density ratio decreases to about 2 when $\delta = 50$. This result shows that data fusion is effective in the scenarios with low SNRs. When $\delta = 5,000$, the disc model suffices. These results are consistent with the analysis in Sect. 5.1.3.

The third set of simulations evaluate the asymptotic relationship between ρ_d and ρ_f when the fusion range is optimized. In Fig. 16, the X- and Y-axis of each data point represent the required network densities for achieving the same coverage that approaches to one under the disc and fusion models, respectively. Note that the Y-axis is plotted in square root scale. The optimal fusion range R_{opt} plotted in

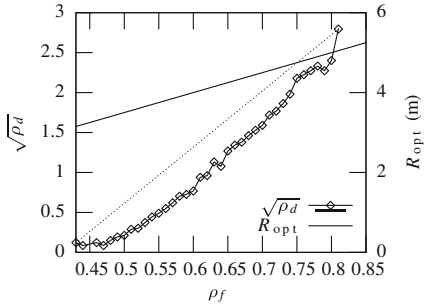


Fig. 16 $\sqrt{\rho_d}$ versus ρ_f of random networks with optimal fusion range R_{opt} ($\delta = 4$)

Fig. 16 is computed for each given ρ_f by numerically maximizing (7). We can see from Fig. 16 that the relationship between $\sqrt{\rho_d}$ and ρ_f is convex and therefore conforms to the theoretical result $\rho_f = \mathcal{O}(\sqrt{\rho_d})$ according to Theorem 2. Moreover, R_{opt} increases with ρ_f, which is also consistent with the analysis in Sect. 4.2.1.

7.2.3 Temporal Coverage

We first evaluate the analytical result on the α-delay of fusion-based detection. Figure 17 plots the α-delay versus the network density. The curve labeled with "analytical" plots the α-delay computed according to Lemma 6 and Eq. (12). The data points labeled with "SIM(ϵ)" represent the simulation results with a constant localization error ϵ. From the figure, we can see that the α-delay decreases with the network density. The simulation result without localization error (i.e., $\epsilon = 0$) confirms the analytical result when the network density is greater than 0.01. When ρ is smaller than 0.01, the simulation result starts to deviate from the analytical result. This is due to the approximation made in the derivation of P_D in Sect. 4.2.2. However, we can see that the maximum error between the analytical and simulation results falls within one detection period. Figure 17 also shows that the impact of localization error is small. The simulation result has a considerable deviation from the analytical result only when the localization error is equal to the fusion range (25 m). In such a case, the target falls completely outside of the fusion range. Moreover, the impact of localization error diminishes as the network density increases. This result demonstrates the robustness of our analysis with respect to localization error, especially in achieving instant detection.

The second set of simulations evaluate the impact of overlap/no-overlap condition on the α-delay under the disc and fusion models, respectively. Figure 18a plots the α-delay versus the network density under the value fusion model. The curves labeled with "analytical (no-overlap)" and "upper bound" plot the α-delay under the no-overlap case (given by Lemma 6) and its upper bound (given by Lemma 9), respectively. We can see that the two analytical results are very close. The other two curves plot the simulation results for the overlap and no-overlap cases, respectively. The simulation results closely match the analytical results when the network density

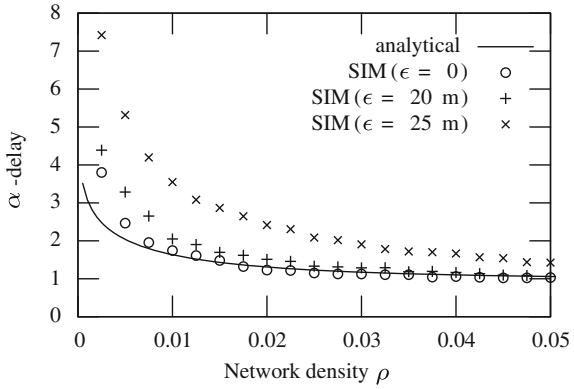


Fig. 17 α-delay versus network density under data fusion model

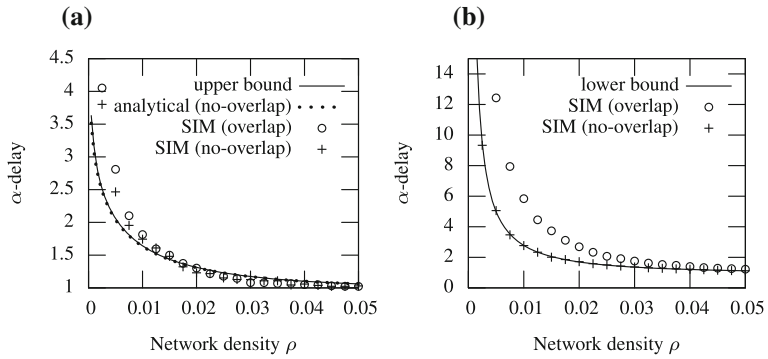


Fig. 18 α-delay versus network density. **a** Data fusion model. **b** Disc model

is greater than 0.02. When ρ is smaller than 0.01, the deviation between the analytical and simulation results is due to the approximation made in the derivation of P_D. Moreover, we can see from Fig. 18a that the overlap/no-overlap condition has little impact on the α-delay under the fusion model. Figure 18b plots the α-delay under the disc model. Note that the lower bound given by Lemma 8 is also the analytical result of α-delay under the no-overlap case given by Lemma 2. We can see that the simulation results confirm the analytical results under the disc model. Moreover, the α-delay significantly increases under the overlap case. Hence, the overlap/no-overlap condition has significant impact on the α-delay under the disc model.

We now evaluate the impact of false alarm rate and SNR on the density ratio. Figure 19a plots the ratio of network densities required by the data fusion and disc models to achieve the same α-delay given various false alarm rates. We can see from Fig. 19a that the disc model requires more than twice sensors when the α-delay approaches to one. Both for the value and decision fusion models, the density ratio decreases if a lower α is required, which is consistent with Theorems 4 and 5. More-

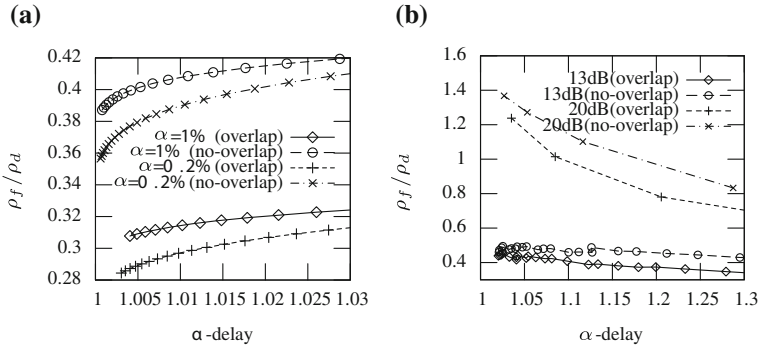
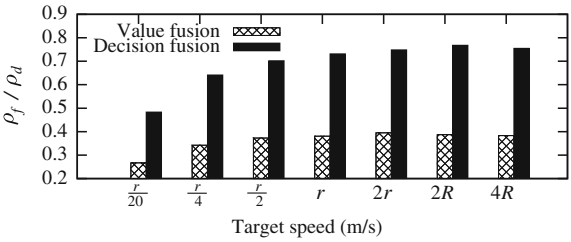


Fig. 19 Density ratio versus α-delay. **a** Given different α (SNR $=$ 10 dB). **b** Given different SNRs ($\alpha = 1\,\%$)

Fig. 20 Density ratio versus target speed (SNR $=$ 13 dB, $\alpha = 5\,\%$, $\tau = 1.05$, $r = 2.25$ m, $R = 8$ m, $T = 1$ s)



over, from the two figures, we can see that the density ratio under the overlap case is smaller than that under the no-overlap case. This is consistent with our observation in the previous set of simulations, i.e., the overlap condition has little impact on the fusion model while leads to significant increase of α-delay under the disc model. Figure 19b plots the ratio of network densities required by the data fusion and disc models given various SNRs. From Fig. 19b, we can see that the density ratio increases with SNR, which is consistent with Theorems 4 and 5. For instance, if the SNR is 13 dB, the density ratio ρ_f/ρ_d is about 0.5 when the α-delay reduces to one. However, if the SNR increases to 20 dB, ρ_f/ρ_d is greater than 1.2 and hence the disc model requires fewer sensors than the fusion model.

As target speed is an important factor of the overlap/no-overlap condition, we finally evaluate its impact on the density ratio. Figure 20 shows the density ratio versus the target speed. We can see that the density ratio significantly increases when the target speed increases from $\frac{r}{20}$ to $2r$. This is due to the significant impact of overlap condition on the disc model, as observed in Fig. 18. Hence, the data fusion models are more robust than the disc model in detecting slowly moving targets.

8 Conclusion

Spatiotemporal coverage is an important performance requirement of many critical sensor network applications. In this paper, we explore the fundamental limits of spatiotemporal coverage based on stochastic data fusion models that jointly process noisy measurements of sensors. The scaling laws between spatiotemporal coverage, network density, and SNR are derived. Data fusion is shown to significantly improve spatiotemporal coverage by exploiting the collaboration among sensors. Our results help understand the limitations of the existing analytical results based on the disc model and provide key insights into the design and analysis of WSNs that adopt data fusion algorithms. Our analyses are verified through simulations based on both synthetic data sets and data traces collected in a real deployment for vehicle detection.

Appendix 1: Proof of Lemma 2

Proof As shown in [29], when the sensors are deployed according to the Poisson process, the probability that there is at least one sensor in a target disc is $p = 1 - \mathrm{e}^{-\rho\pi r^2}$. Suppose the target is detected in the Jth ($J \geq 1$) detection period. As there is no overlap between any two target discs, the unit detections are independent from each other. Therefore, J follows the geometric distribution with a success probability of p in each Bernoulli trial (i.e., each unit detection). Moreover, according to the definition of r in (4), the false alarm rate in each unit detection is no greater than α. According to Definition 3, the α-delay is given by $\tau = \mathbb{E}[J] = \frac{1}{p} = \frac{1}{1-\mathrm{e}^{-\rho\pi r^2}}$.

Appendix 2: Proof of Lemma 3

Proof We first discuss the necessary and sufficient condition that p is (α, β)-covered. When no target is present, all sensors measure *i.i.d.* noises and hence $Y|H_0 = \sum_{i\in\mathbf{F}(p)} n_i \sim \mathcal{N}(\mu N(p), \sigma^2 N(p))$. Therefore, the false alarm rate is $P_F = \mathbb{P}(Y \geq T|H_0) = Q\left(\frac{T-\mu N(p)}{\sigma\sqrt{N(p)}}\right)$, where T is the detection threshold. As P_D is a non-decreasing function of P_F [44], it is maximized when P_F is set to be the upper bound α. Such a scheme is referred to as the constant false alarm rate detector [44]. Let $P_F = \alpha$, the optimal detection threshold can be derived as $T_{\text{opt}} = \mu N(p) + \sigma Q^{-1}(\alpha)\sqrt{N(p)}$. When the target is present, we have

$$Y|H_1 = \sum_{i\in\mathbf{F}(p)} s_i + n_i \sim \mathcal{N}(\mu N(p) + \sum_{i\in\mathbf{F}(p)} s_i, \sigma^2 N(p)).$$

Therefore, the detection probability at p is given by

$$P_D(p) = \mathbb{P}(Y \geq T | H_1) = Q\left(\frac{T - \mu N(p) - \sum_{i \in \mathbf{F}(p)} s_i}{\sigma\sqrt{N(p)}}\right).$$

By replacing T with T_{opt} and solving $P_D(p) \geq \beta$, we have the necessary and sufficient condition that p is (α, β)-covered:

$$\frac{\sum_{i \in \mathbf{F}(p)} s_i}{\sqrt{N(p)}} \geq \sigma\left(Q^{-1}(\alpha) - Q^{-1}(\beta)\right). \tag{19}$$

As the random network is stationary, the fraction of covered area equals the probability that an arbitrary point is covered by the network [29]. Therefore, the spatial coverage of the network is given by (6). □

Appendix 3: Proof of Lemma 4

Proof We first prove that the $\{s_i | i \in \mathbf{F}(p)\}$ are *i.i.d.* for given p and derive the formulas for μ_s and σ_s^2. As sensors are deployed uniformly and independently, $\{d_i | i \in \mathbf{F}(p)\}$ are *i.i.d.* for given p, where d_i is the distance between sensor i and point p. To simplify our discussion, we now temporarily assume that there is no localization error, i.e., $\epsilon = 0$. Therefore, $\{s_i | i \in \mathbf{F}(p)\}$ are *i.i.d.* for given p, as s_i is a function of d_i. Suppose the coordinates of point p and sensor i are (x_p, y_p) and (x_i, y_i), respectively. The posterior PDF of (x_i, y_i) is $f(x_i, y_i) = \frac{1}{\pi R^2}$ where $(x_i - x_p)^2 + (y_i - y_p)^2 \leq R^2$. Hence, the posterior CDF of d_i is given by $F(d_i) = \int_0^{2\pi} \mathrm{d}\theta \int_0^{d_i} \frac{1}{\pi R^2} \cdot x \mathrm{d}x = \frac{d_i^2}{R^2}$ where $d_i \in [0, R]$. Therefore, we have

$$\mu_s = \int_0^R s_i \mathrm{d}F(d_i) = \frac{2S_0}{R^2} \cdot \int_0^R x w(x) \mathrm{d}x, \tag{20}$$

$$\sigma_s^2 = \int_0^R s_i^2 \mathrm{d}F(d_i) - \mu_s^2 = \frac{2S_0^2}{R^2} \int_0^R x w^2(x) \mathrm{d}x - \mu_s^2. \tag{21}$$

By letting $\mu_0 = \frac{2}{R^2} \int_0^R x w(x) \mathrm{d}x$ and $\sigma_0^2 = \frac{2}{R^2} \int_0^R x w^2(x) \mathrm{d}x - \mu_0^2$, we have $\mu_s = S_0 \mu_0$ and $\sigma_s^2 = S_0^2 \sigma_0^2$.

A straightforward approximation is to replace $\sum_{i \in \mathbf{F}(p)} s_i$ in (6) with its mean $\mu_s N(p)$. However, doing so ignores the distribution of $\sum_{i \in \mathbf{F}(p)} s_i$. As $N(p)$ follows the Poisson distribution, $\sum_{i \in \mathbf{F}(p)} s_i$ follows the *compound Poisson distribution*, which has no closed-form PDF and CDF. We approximate the compound Poisson distribution using the normal distribution. The intuition behind this approximation is the CLT by assuming $N(p)$ is a constant. Therefore, $\sum_{i \in \mathbf{F}(p)} s_i \sim \mathcal{N}(\mu_s N(p), \sigma_s^2 N(p))$. When the target is present, $Y | H_1 = \sum_{i \in \mathbf{F}(p)} s_i + \sum_{i \in \mathbf{F}(p)} n_i$. As the sum of two independent Gaussians is also Gaussian, $Y | H_1$ follows the normal distribution, i.e., $Y | H_1 \sim \mathcal{N}(\mu_s N(p) + \mu N(p), \sigma_s^2 N(p) + \sigma^2 N(p))$. There-

fore, the detection probability at point p is given by $P_D(p) = \mathbb{P}(Y \geq T|H_1) \simeq Q\left(\frac{T-\mu_s N(p)-\mu N(p)}{\sqrt{\sigma_s^2+\sigma^2}\cdot\sqrt{N(p)}}\right)$. By replacing T with the optimal detection threshold T_{opt} (derived in the proof of Lemma 3) and solving $P_D(p) \geq \beta$, the condition that p is (α, β)-covered is given by $N(p) \geq \gamma(R)$. The approximate formula of spatial coverage is then given by

$$c \simeq \mathbb{P}(N(p) \geq \gamma(R)) = 1 - F_{\text{Poi}}(\gamma(R)|\rho\pi R^2), \tag{22}$$

where $F_{\text{Poi}}(\cdot|\lambda)$ is the CDF of the Poisson distribution $\text{Poi}(\lambda)$. When $\rho\pi R^2$ is large enough, the Poisson distribution $\text{Poi}(\rho\pi R^2)$ can be excellently approximated by the normal distribution $\mathcal{N}(\rho\pi R^2, \rho\pi R^2)$. Therefore, Eq. (22) can be further approximated by (7). □

Appendix 4: Proof of Lemma 5

Proof For any point p, $\sum_{i\in\mathbf{F}(p)} s_i \geq S_0 \cdot w(R+\epsilon)\cdot N(p)$, as $s_i \geq S_0 \cdot w(R+\epsilon)$ for any sensor i in $\mathbf{F}(p)$. If $\frac{S_0\cdot w(R+\epsilon)\cdot N(p)}{\sqrt{N(p)}} \geq \sigma\left(Q^{-1}(\alpha) - Q^{-1}(\beta)\right)$, Eq. (19) must hold. Therefore, by solving $N(p)$, the sufficient condition that p is (α, β)-covered is $N(p) \geq \Gamma(R)$. Moreover, as $N(p) \sim \text{Poi}(\rho\pi R^2)$, we have

$$c = \mathbb{P}(\text{point } p \text{ is } (\alpha,\beta) - \text{covered}) \geq \mathbb{P}(N \geq \Gamma(R)) = 1 - F_{\text{Poi}}(\Gamma(R)|\rho\pi R^2).$$

Therefore, the lower bound of c is given by (8). When $\rho\pi R^2$ is large enough, the normal distribution $\mathcal{N}(\rho\pi R^2, \rho\pi R^2)$ excellently approximates the Poisson distribution $\text{Poi}(\rho\pi R^2)$. Therefore, Eq. (8) can be approximated by (10). □

Appendix 5: Proof of Lemma 6

Proof Denote A_j as the event that the target is not detected in the jth unit detection. Thus, the probability of A_j is $\mathbb{P}(A_j) = 1 - P_{Dj}$. Suppose the target is detected in the Jth unit detection. Although the intrusion detection is a series of infinite Bernoulli trials, J does not follow the geometric distribution because the success probability of each Bernoulli trial (i.e., P_{Dj}) is a random variable rather than a constant. The mean of J is given by

$$\mathbb{E}[J] = 1 \cdot \mathbb{P}(\bar{A}_1) + \sum_{j=2}^{\infty} j \cdot \mathbb{P}\left(\bigcap_{k=1}^{j-1} A_k \bigcap \bar{A}_j\right) \tag{23}$$

$$= 1 - \mathbb{P}(A_1) + \sum_{j=2}^{\infty} j \cdot \left(\mathbb{P}\left(\bigcap_{k=1}^{j-1} A_k\right) - \mathbb{P}\left(\bigcap_{k=1}^{j} A_k\right)\right)$$

$$= 1 + \sum_{j=1}^{\infty} \mathbb{P}\left(\bigcap_{k=1}^{j} A_k\right) \tag{24}$$

$$= 1 + \sum_{j=1}^{\infty} \prod_{k=1}^{j} \mathbb{P}(A_k) \tag{25}$$

$$= 1 + \sum_{j=1}^{\infty} \prod_{k=1}^{j} (1 - P_{Dk}). \tag{26}$$

Note that the $\bigcap_{k=1}^{j-1} A_k \bigcap \bar{A}_j$ in (23) represents the event that the target is not detected from the first to the $(j-1)$th unit detection but detected in the jth unit detection. As the measurements in different sampling intervals are mutually independent, $\{A_j | j \geq 1\}$ are mutually independent. Hence, Eq. (25) follows. We now explain the physical meaning of $\mathbb{E}[J]$. For a given randomly deployed network, if the target always appears at a fixed location and travels a fixed trajectory, according to (11), $\{P_{Dj} | j \geq 1\}$ are fixed values as $\{N_j | j \geq 1\}$ are fixed. As each unit detection is probabilistic, the $\mathbb{E}[J]$ is the average delay of detecting the target with fixed trajectory. For the target that appears at random location and travels arbitrary trajectory, $\{P_{Dj} | j \geq 1\}$ are random variables as $\{N_j | j \geq 1\}$ are random variables. Therefore, the average delay for detecting the target with arbitrary trajectory, i.e., α-delay, is given by $\tau = \mathbb{E}[\mathbb{E}[J]]$, where $\mathbb{E}[\mathbb{E}[J]]$ is the average of $\mathbb{E}[J]$ taken over all possible target trajectories. If fusion ranges do not overlap, $\{N_j | j \geq 1\}$ are *i.i.d.* random variables. Hence, $\{P_{Dj} | j \geq 1\}$ are also *i.i.d.* random variables. Therefore,

$$\tau = \mathbb{E}[\mathbb{E}[J]] = 1 + \sum_{j=1}^{\infty} \prod_{k=1}^{j} \mathbb{E}[1 - P_{Dk}]$$

$$= 1 + \sum_{j=1}^{\infty} (1 - \mathbb{E}[P_D])^j = \frac{1}{\mathbb{E}[P_D]}. \qquad \square$$

Appendix 6: Proof of Theorem 1

Proof As ρ_f is large to provide a high level of spatial coverage under the fusion model, the lower bound of spatial coverage, c_L, is given by (10) according to Lemma 5. We define $h_1(\rho_f) = \frac{\Gamma(R)}{\sqrt{\pi}R} \cdot \frac{1}{\sqrt{\rho_f}}$, $h_2(\rho_f) = \sqrt{\pi}R \cdot \sqrt{\rho_f}$ and hence $c_L = Q(h_1(\rho_f) - h_2(\rho_f))$. When $\rho_f \to \infty$, $h_2(\rho_f)$ dominates $h_1(\rho_f)$ as $\lim_{\rho_f \to \infty} \frac{h_1(\rho_f)}{h_2(\rho_f)} = 0$. Hence, $c \geq c_L = Q(-h_2(\rho_f)) = Q(-\sqrt{\pi}R \cdot \sqrt{\rho_f})$ when $\rho_f \to \infty$. Define $x = Q^{-1}(c)$. We have $\rho_f \leq \frac{1}{\pi R^2} x^2$ when $c \to 1^-$.

Under the disc model, by replacing $c = Q(x) = 1 - \Phi(x)$ in (5) and solving ρ_d, we have $\rho_d = -\frac{1}{\pi r^2} \ln \Phi(x)$, where $\Phi(x)$ is the CDF of the standard normal distribution. Hence, we have

$$\lim_{c \to 1^-} \frac{\rho_f}{\rho_d} \leq \lim_{x \to -\infty} \frac{\frac{1}{\pi R^2} x^2}{-\frac{1}{\pi r^2} \ln \Phi(x)} = -\frac{r^2}{R^2} \lim_{x \to -\infty} \frac{x^2}{\ln \Phi(x)}.$$

As $\lim_{x \to -\infty} \frac{x^2}{\ln \Phi(x)} = -2$ [49], we have $\lim_{c \to 1^-} \frac{\rho_f}{\rho_d} \leq \frac{2r^2}{R^2}$. Therefore, the asymptotic upper bound of ρ_f is given by (13). □

Appendix 7: Proof of Theorem 2

Proof We choose R by

$$\frac{\xi}{\pi} \cdot \frac{\Gamma(R)}{R^2} = \rho_f, \tag{27}$$

where ξ is a constant and $\xi > 1$. It is easy to verify that the chosen R is order-optimal for the lower bound of coverage (i.e., c_L). Moreover, it is easy to verify that both the chosen R and $\Gamma(R)$ increase with ρ_f. By replacing ρ_f in (10) with (27), c_L is given by $c_L = Q\left(\left(\frac{1}{\sqrt{\xi}} - \sqrt{\xi}\right) \cdot \sqrt{\Gamma(R)}\right) = 1 - \Phi(\eta z)$, where $\eta = \frac{1}{\sqrt{\xi}} - \sqrt{\xi}$ is a constant and $z = \sqrt{\Gamma(R)}$. Hence we have $c \geq c_L = 1 - \Phi(\eta z)$. According to (5), the network density under the disc model satisfies $\rho_d = -\frac{1}{\pi r^2} \ln(1 - c) \geq -\frac{1}{\pi r^2} \ln \Phi(\eta z)$. Hence, the ratio ρ_f^b / ρ_d, where b is a positive constant, satisfies

$$\begin{aligned}
\lim_{c \to 1^-} \frac{\rho_f^b}{\rho_d} &\leq \lim_{R \to \infty} \frac{\left(\frac{\xi}{\pi}\right)^b \cdot \frac{\Gamma^b(R)}{R^{2b}}}{-\frac{1}{\pi r^2} \ln \Phi(\eta z)} \\
&= -\frac{\xi^b r^2}{\pi^{b-1}} \cdot \lim_{z \to \infty} \frac{z^2}{\ln \Phi(\eta z)} \cdot \lim_{R \to \infty} \frac{\Gamma^{b-1}(R)}{R^{2b}} \\
&= \frac{2\xi^b r^2}{\pi^{b-1} \eta^2} \cdot \lim_{R \to \infty} \frac{\Gamma^{b-1}(R)}{R^{2b}}.
\end{aligned}$$

Note that $\lim_{z\to\infty} \frac{z^2}{\ln \Phi(\eta z)} = -\frac{2}{\eta^2}$ [49] in the above derivation. As $w(x) = \Theta(x^{-k})$ and ϵ is constant, $\Gamma(R) = \Theta(1/w^2(R+\epsilon)) = \Theta((R+\epsilon)^{2k}) = \Theta(R^{2k})$ and hence $\Gamma^{b-1}(R) = \Theta(R^{2kb-2k})$. Therefore, $\lim_{R\to\infty} \frac{\Gamma^{b-1}(R)}{R^{2b}} = \lim_{R\to\infty} R^{2kb-2k-2b}$. If $b \le \frac{k}{k-1}$, $\lim_{R\to\infty} \frac{\Gamma^{b-1}(R)}{R^{2b}}$ is a constant and hence $\lim_{c\to 1^-} \frac{\rho_f^b}{\rho_d}$ is upper-bounded by a constant. Hence, we have (14). We note that although the chosen R is not optimal for c, the upper bound given by (14) still holds if R is optimal for c. □

Appendix 8: Proof of Lemma 7

Proof We abuse the symbols a bit to use N instead of N_j and P_D instead of P_{Dj} as we are not interested in the index of unit detection. As $\rho \to \infty$, $N \to \infty$ almost surely. In (11), the second item $-\frac{\mu_s}{\sqrt{\sigma_s^2+\sigma^2}} \cdot \sqrt{N}$ dominates when $\rho \to \infty$, since the first item $\frac{\sigma}{\sqrt{\sigma_s^2+\sigma^2}} \cdot Q^{-1}(\alpha)$ is a constant. Therefore, it's safe to use $P_D = Q(\gamma\sqrt{N})$ to approximate (11), where $\gamma = -\frac{\mu_s}{\sqrt{\sigma_s^2+\sigma^2}}$. From Lemma 2 and 6, if the same α-delay of τ is achieved under the two models, we have

$$\mathbb{E}[P_D] = 1 - \mathrm{e}^{-\rho_d \pi r^2}. \tag{28}$$

We first prove the lower bound in (16). It is easy to verify that $P_D = Q(\gamma\sqrt{N})$ is a concave function. According to Jensen's inequality, we have $\mathbb{E}[P_D] \le Q(\gamma\sqrt{\mathbb{E}[N]}) = Q(\gamma\sqrt{\rho_f \pi R^2})$. From (28), we have $1-\mathrm{e}^{-\rho_d \pi r^2} = \mathbb{E}[P_D] \le Q(\gamma\sqrt{\rho_f \pi R^2})$. Accordingly, $\rho_d \le -\frac{1}{\pi r^2} \ln \Phi(\gamma\sqrt{\pi} R \cdot \sqrt{\rho_f})$, where $\Phi(x) = 1 - Q(x)$. Hence, the density ratio satisfies

$$\lim_{\tau\to 1^+} \frac{\rho_f}{\rho_d} \ge -\pi r^2 \cdot \lim_{\rho_f\to\infty} \frac{\rho_f}{\ln \Phi(\gamma\sqrt{\pi} R \cdot \sqrt{\rho_f})} = \frac{2}{\gamma^2 R^2} \cdot r^2.$$

In the above derivation, we use the equality $\lim_{x\to\infty} \frac{x}{\ln \Phi(\eta\sqrt{x})} = -\frac{2}{\eta^2}$, which has been proved in [49].

We now prove the upper bound in (16). As $P_D > 0$, according to Markov's inequality, for any given number c, we have

$$\mathbb{E}[P_D] \ge c \cdot \mathbb{P}(P_D \ge c). \tag{29}$$

We define ξ and c as follows:

$$\xi = \frac{\gamma^2 + 2 - \sqrt{\gamma^4 + 4\gamma^2}}{2}, \quad c = Q(\gamma\sqrt{\xi \rho_f \pi R^2}). \tag{30}$$

It's easy to verify that $\xi \in (0, 1)$. Therefore,

$$\mathbb{P}(P_D \geq c) = \mathbb{P}\left(Q(\gamma\sqrt{N}) \geq Q(\gamma\sqrt{\xi\rho_f\pi R^2})\right) = \mathbb{P}(N \geq \xi\rho_f\pi R^2).$$

As $N \sim \text{Poi}(\rho_f\pi R^2)$ and the Poisson distribution approaches the normal distribution $\mathcal{N}(\rho_f\pi R^2, \rho_f\pi R^2)$ when $\rho_f \to \infty$, we have

$$\mathbb{P}(P_D \geq c) = Q\left(\frac{\xi\rho_f\pi R^2 - \rho_f\pi R^2}{\sqrt{\rho_f\pi R^2}}\right) = Q\left((\xi-1)\sqrt{\rho_f\pi R^2}\right).$$

By replacing c and $\mathbb{P}(P_D \geq c)$ in (29), we have

$$\mathbb{E}[P_D] \geq Q\left(\gamma\sqrt{\xi\rho_f\pi R^2}\right) \cdot Q\left((\xi-1)\sqrt{\rho_f\pi R^2}\right).$$

It is easy to verify that $\gamma\sqrt{\xi} = \xi - 1$. Thus the above inequality reduces to $\mathbb{E}[P_D] \geq Q^2(h\sqrt{\rho_f})$, where $h = \gamma\sqrt{\xi\pi}R$. From (28), we have $1 - \mathrm{e}^{-\rho_d\pi r^2} = \mathbb{E}[P_D] \geq Q^2(h\sqrt{\rho_f})$. Accordingly, $\rho_d \geq -\frac{1}{\pi r^2}\cdot(\ln(1+Q(h\sqrt{\rho_f})) + \ln\Phi(h\sqrt{\rho_f}))$. Hence,we have

$$\begin{aligned}\lim_{\tau\to 1^+}\frac{\rho_f}{\rho_d} &\leq -\pi r^2 \lim_{\rho_f\to\infty}\frac{\rho_f}{\ln(1+Q(h\sqrt{\rho_f})) + \ln\Phi(h\sqrt{\rho_f})} \\ &= -\pi r^2 \lim_{\rho_f\to\infty}\frac{\rho_f}{\ln\Phi(h\sqrt{\rho_f})} = \frac{2}{\xi\gamma^2R^2}\cdot r^2. \end{aligned} \tag{31}$$

Note that $h = \gamma\sqrt{\xi\pi}R < 0$ and $\ln(1 + Q(h\sqrt{\rho_f})) = \ln 2$ when $\rho_f \to \infty$. We also use the aforementioned equality $\lim_{x\to\infty}\frac{x}{\ln\Phi(\eta\sqrt{x})} = -\frac{2}{\eta^2}$ [49] to derive (31). □

Appendix 9: Proof of Theorem 4

Proof In Lemma 7, γ depends on the PSNR δ, i.e.,

$$\gamma = -\frac{\mu_s}{\sqrt{\sigma_s^2+\sigma^2}} = -\frac{S_0\mu_0}{\sqrt{S_0^2\sigma_0^2+\sigma^2}} = -\frac{\mu_0}{\sqrt{\sigma_0^2+\frac{1}{\delta^2}}},$$

where μ_0 and σ_0^2 (both defined in the proof of Lemma 4) are constants. Moreover, ξ is a function of γ (given by (30)). Accordingly, γ and ξ are both constants when δ is fixed or approaches infinity. Hence, according to Lemma 7, the tight bound of

the density ratio is $\lim_{\tau\to 1^+} \rho_f/\rho_d = \Theta(r^2)$. As $w^{-1}(x) = \Theta(x^{-1/k})$, according to (4), $r^2 = \Theta\left(\left(\frac{\delta}{Q^{-1}(\alpha)}\right)^{2/k}\right)$ for fixed β. Therefore, we have (17). □

Appendix 10: Proof of Lemma 8

Proof Let A_j denote the event that the target is not detected in the jth unit detection and C_j denote the corresponding target disc. Suppose the target is detected in the Jth unit detection. Recall (24), we have $\mathbb{E}[J] = 1 + \sum_{j=1}^{\infty} \mathbb{P}\left(\bigcap_{k=1}^{j} A_k\right) = 1 + \sum_{j=1}^{\infty} \prod_{k=1}^{j} \mathbb{P}\left(A_k \left| \bigcap_{l=1}^{k-1} A_l \right.\right)$. The above derivation follows the definition of conditional probability. Let C denote the common area between the kth target disc and the union of all the previous target discs, i.e., $C = C_k \cap (\bigcup_{l=1}^{k-1} C_l)$. Therefore, $C \geq 0$ and

$$\mathbb{P}\left(A_k \left| \bigcap_{l=1}^{k-1} A_l \right.\right) = \mathbb{P}\left(\text{there is no sensor in } (C_k - C)\right) = \mathrm{e}^{-\rho(\pi r^2 - C)} \geq \mathrm{e}^{-\rho\pi r^2}.$$

Hence, $\tau = \mathbb{E}[J] \geq 1 + \sum_{j=1}^{\infty} \left(\mathrm{e}^{-\rho\pi r^2}\right)^j = \frac{1}{1-\mathrm{e}^{-\rho\pi r^2}}$. □

Appendix 11: Proof of Lemma 9

Proof We first introduce the generalized Hölder's inequality [15]. *For random variables* X_i, $i = 1, \ldots, n$, *we have* $\mathbb{E}\left[\prod_{i=1}^{n} |X_i|\right] \leq \prod_{i=1}^{n} \left(\mathbb{E}\left[|X_i|^{p_i}\right]\right)^{1/p_i}$ *where* $p_i > 1$ *and* $\sum_{i=1}^{n} p_i^{-1} = 1$. If X_i, $i = 1, \ldots, n$, are identically distributed, by setting $p_i = n$, we have

$$\mathbb{E}\left[\prod_{i=1}^{n} |X_i|\right] \leq \mathbb{E}\left[|X|^n\right], \tag{32}$$

where X can be any X_i. In our problem, $\{N_j | j \geq 1\}$ are identically distributed random variables due to the Poisson process. As P_{Dj} is a function of N_j (given by (11)), $\{P_{Dj} | j \geq 1\}$ are also identically distributed random variables. Recall (26), by applying the inequality (32) , the α-delay of fusion-based detection can be derived as

$$\tau = \mathbb{E}[\mathbb{E}[J]] = 1 + \sum_{j=1}^{\infty} \mathbb{E}\left[\prod_{k=1}^{j}(1 - P_{Dk})\right]$$
$$\leq 1 + \sum_{j=1}^{\infty} \mathbb{E}[(1 - P_D)^j] = \mathbb{E}\left[\frac{1}{P_D}\right]. \qquad \square$$

Appendix 12: Proof of Theorem 5

Proof According to Lemma 8 and Lemma 9, we have

$$1/(1 - \mathrm{e}^{-\rho_d \pi r^2}) \leq \tau \leq \mathbb{E}\,[1/P_D]\,. \tag{33}$$

We first find an upper bound of $\mathbb{E}\,[1/P_D]$. As discussed in the proof of Lemma 7, it is safe to use $P_D = Q(\gamma\sqrt{N})$ to approximate (11), where $\gamma = -\frac{\mu_s}{\sqrt{\sigma_s^2 + \sigma^2}}$. As $N \sim \mathrm{Poi}(\rho_f \pi R^2)$ and the Poisson distribution approaches to the normal distribution $\mathcal{N}(\rho_f \pi R^2, \rho_f \pi R^2)$ when $\rho_f \to \infty$, for any given constant $\xi \in (0, 1)$, we have $\mathbb{P}(N \geq \xi \rho_f \pi R^2) = Q\left(\frac{\xi \rho_f \pi R^2 - \rho_f \pi R^2}{\sqrt{\rho_f \pi R^2}}\right) = Q\left((\xi - 1)\sqrt{\rho_f \pi R^2}\right)$. When $\rho_f \to \infty$, $\mathbb{P}(N \geq \xi \rho_f \pi R^2) \to 1$, i.e., $N \geq \xi \rho_f \pi R^2$ with high probability. Moreover, as $1/P_D = 1/Q(\gamma N)$ is a decreasing function of N, $\mathbb{E}[1/P_D] \leq 1/Q(\gamma\sqrt{\xi \rho_f \pi R^2})$ with high probability. Furthermore, according to (33), we have $1/(1 - \mathrm{e}^{-\rho_d \pi r^2}) \leq 1/Q(\gamma\sqrt{\xi \rho_f \pi R^2})$ probability when $\rho_f \to \infty$. After manipulation, we have $\rho_d \geq -\frac{1}{\pi r^2} \ln\left(\Phi(\gamma\sqrt{\xi\pi} R \sqrt{\rho_f})\right)$, where $\Phi(x) = 1 - Q(x)$. Hence, we have

$$\lim_{\tau \to 1^+} \frac{\rho_f}{\rho_d} \leq -\pi r^2 \lim_{\rho_f \to \infty} \frac{\rho_f}{\ln\left(\Phi(\gamma\sqrt{\xi\pi} R \sqrt{\rho_f})\right)} = \frac{2}{\gamma^2 \xi R^2} \cdot r^2. \tag{34}$$

In the above derivation, we use the equality $\lim_{x \to \infty} \frac{x}{\ln \Phi(\vartheta\sqrt{x})} = -\frac{2}{\vartheta^2}$ that has been proved in [49]. Hence, the upper bound of the density ratio is $\lim_{\tau \to 1^+} \rho_f/\rho_d = \mathcal{O}(r^2)$. As $r^2 = \Theta\left(\left(\frac{\delta}{Q^{-1}(\alpha)}\right)^{2/k}\right)$, we have (18). $\square$

References

1. N. Ahmed, S.S. Kanhere, S. Jha, Probabilistic coverage in wireless sensor networks, in *The 30th IEEE Conference on Local Computer Networks (LCN)*, Sydney, Australia, pp. 672–681 (2005)
2. H.M. Ammari, S. Das, Integrated coverage and connectivity in wireless sensor networks: A two-dimensional percolation problem. IEEE Trans. Comput. **57**(10), 1423–1434 (2008)
3. N. Bisnik, A. Abouzeid, V. Isler, Stochastic event capture using mobile sensors subject to a quality metric, in *The 12th Annual International Conference on Mobile Computing and Networking (MobiCom)*, Los Angeles, CA, USA, pp. 98–109 (2006)
4. P. Brass, Bounds on coverage and target detection capabilities for models of networks of mobile sensors. ACM Trans. Sens. Netw. **3**(2), 9 (2007)
5. Q. Cao, T. Yan, J. Stankovic, T. Abdelzaher, Analysis of target detection performance for wireless sensor networks, in *The 1st International Conference on Distributed Computing in Sensor Systems (DCOSS)*, Marina del Rey, CA, USA, pp. 276–292 (2005)
6. Z. Chair, P.K. Varshney, Optimal data fusion in multiple sensor detection systems. IEEE Trans. Aerosp. Electron. Syst. **22**(1), 98–101 (1986)
7. K. Chakrabarty, S.S. Iyengar, H. Qi, E. Cho, Grid coverage for surveillance and target location in distributed sensor networks. IEEE Trans. Comput. **51**, 1448–1453 (2002)
8. W.P. Chen, J.C. Hou, L. Sha, Dynamic clustering for acoustic target tracking in wireless sensor networks. IEEE Trans. Mobile Comput. **3**(3), 258–271 (2004)
9. S.Y. Cheung, S. Coleri, B. Dundar, S. Ganesh, C.W. Tan, P. Varaiya, A sensor network for traffic monitoring (plenary talk), in *The 3rd International Symposium on Information Processing in Sensor Networks (IPSN)*, Berkeley, CA, USA (2004)
10. T. Clouqueur, K.K. Saluja, P. Ramanathan, Fault tolerance in collaborative sensor networks for target detection. IEEE Trans. Comput. **53**(3), 320–333 (2004)
11. D. Davis, C. Davis, *Sound System Engineering* (Focal Press, Taylor & Francis Group, Waltham, Massachusetts, 1997)
12. O. Dousse, C. Tavoularis, P. Thiran, Delay of intrusion detection in wireless sensor networks, in *The 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, Florence, Italy, pp. 155–165 (2006)
13. M. Duarte, Y.H. Hu, Distance based decision fusion in a distributed wireless sensor network, in *The 2nd International Workshop on Information Processing in Sensor Networks (IPSN)*, Palo Alto, CA, USA, pp. 392–404 (2003)
14. M. Duarte, Y.H. Hu, Vehicle classification in distributed sensor networks. J. Parallel Distrib. Comput. **64**(7), 826–838 (2004)
15. H. Finner, A generalization of Hölder's inequality and some probability inequalities. Ann. Probab. **20**(4), 1893–1901 (1992)
16. B.P. Flanagan, K.W. Parker, Robust distributed detection using low power acoustic sensors. Technical report, The MITRE Corporation (2005). http://www.mitre.org/work/tech_papers/tech_papers_05/05_0329/
17. L. Gu, D. Jia, P. Vicaire, T. Yan, L. Luo, A. Tirumala, Q. Cao, T. He, J.A. Stankovic, T. Abdelzaher, B.H. Krogh, Lightweight detection and classification for wireless sensor networks in realistic environments, in *The 3rd ACM Conference on Embedded Networked Sensor Systems (SenSys)*, San Diego, CA, USA, pp. 205–217 (2005)
18. C. Gui, P. Mohapatra, Power conservation and quality of surveillance in target tracking sensor networks, in *The 10th Annual International Conference on Mobile Computing and Networking (MobiCom)*, Philadelphia, PA, USA, pp. 129–143 (2004)
19. M. Hata, Empirical formula for propagation loss in land mobile radio services. IEEE Trans. Veh. Technol. **29**(3), 317–325 (1980)
20. T. He, S. Krishnamurthy, J.A. Stankovic, T. Abdelzaher, L. Luo, R. Stoleru, T. Yan, L. Gu, J. Hui, B. Krogh, Energy-efficient surveillance system using wireless sensor networks, in *The 2nd International Conference on Mobile Systems, Applications, and Services (MobiSys)*, Boston, MA, USA, pp. 270–283 (2004)

21. M. Hefeeda, H. Ahmadi, A probabilistic coverage protocol for wireless sensor networks, in *The 15th IEEE International Conference on Network Protocols (ICNP)*, Beijing, China, pp. 41–50 (2007)
22. S. Kumar, T.H. Lai, A. Arora, Barrier coverage with wireless sensors, in *The 11th Annual International Conference on Mobile Computing and Networking (MobiCom)*, Cologne, Germany, pp. 284–298 (2005)
23. S. Kumar, T.H. Lai, J. Balogh, On k-coverage in a mostly sleeping sensor network, in *The 10th Annual International Conference on Mobile Computing and Networking (MobiCom)*, Philadelphia, PA, USA, pp. 144–158 (2004)
24. L. Lazos, R. Poovendran, J.A. Ritcey, Probabilistic detection of mobile targets in heterogeneous sensor networks, in *The 6th International Symposium on Information Processing in Sensor Networks (IPSN)*, Cambridge, MA, USA, pp. 519–528 (2007)
25. D. Li, Y.H. Hu, Energy-based collaborative source localization using acoustic micro-sensor array. EUROSIP J. Appl. Signal Proces. **2003**(4), 321–337 (2003)
26. D. Li, K. Wong, Y.H. Hu, A. Sayeed, Detection, classification and tracking of targets in distributed sensor networks. IEEE Signal Proces. Mag. **19**(2), 17–29 (2002)
27. X.Y. Li, P.J. Wan, O. Frieder, Coverage in wireless Ad hoc sensor networks. IEEE Trans. Comput. **52**(6), 753–763 (2003)
28. B. Liu, O. Dousse, J. Wang, A. Saipulla, Strong barrier coverage of wireless sensor networks, in *The 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, Hong Kong SAR, China, pp. 284–298 (2008)
29. B. Liu, D. Towsley, A study on the coverage of large-scale sensor networks, in *The 1st IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS)*, Fort Lauderdale, FL, USA, pp. 475–483(2004)
30. A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, J. Anderson, Wireless sensor networks for habitat monitoring, in *The 1st ACM International Workshop on Wireless Sensor Networks and Applications (WSNA)*, Atlanta, GA, USA, pp. 88–97 (2002)
31. S. Meguerdichian, F. Koushanfar, M. Potkonjak, M.B. Srivastava, Coverage problems in wireless ad-hoc sensor networks, in *The 20th IEEE International Conference on Computer Communications (INFOCOM)*, Anchorage, AK, USA, pp. 1380–1387 (2001)
32. S. Meguerdichian, F. Koushanfar, G. Qu, M. Potkonjak, Exposure in wireless ad-hoc sensor networks, in *The 7th Annual International Conference on Mobile Computing and Networking (MobiCom)*, Rome, Italy, pp. 139–150 (2001)
33. NIST/SEMATECH: e-Handbook of Statistical Methods. The National Institute of Standards and Technology (NIST), Information Technology Laboratory, Statistical Engineering Division (2011). http://www.itl.nist.gov/div898/handbook/
34. R. Niu, P.K. Varshney, Distributed detection and fusion in a large wireless sensor network of random size. EURASIP J. Wireless Commun. Netw. **2005**(4), 462–472 (2005)
35. A. Nordio, C. Chiasserini, E. Viterbo, Quality of field reconstruction in sensor networks, in *The 26th IEEE International Conference on Computer Communications (INFOCOM)*, Anchorage, AK, USA, pp. 2406–2410 (2007)
36. A. Nordio, C. Chiasserini, E. Viterbo, The impact of quasi-equally spaced sensor layouts on field reconstruction, in *The 6th International Symposium on Information Processing in Sensor Networks (IPSN)*, Cambridge, MA, USA, pp. 274–282 (2007)
37. S. Ren, Q. Li, H. Wang, X. Chen, X. Zhang, Design and analysis of sensing scheduling algorithms under partial coverage for object detection in sensor networks. IEEE Trans. Parallel Distrib. Syst. **18**(3), 334–350 (2007)
38. S. Shakkottai, R. Srikant, N.B. Shroff, Unreliable sensor grids: coverage, connectivity and diameter, in *The 22nd IEEE International Conference on Computer Communications (INFOCOM)*, San Franciso, CA, USA, pp. 1073–1083 (2003)
39. X. Sheng, Y.H. Hu, Energy based acoustic source localization, in *The 2nd International Workshop on Information Processing in Sensor Networks (IPSN)*, Palo Alto, CA, USA, pp. 551–551 (2003)

40. D.W. Stroock, S.S. Varadhan,*Multidimensional Diffusion Processes*, vol. 233 (Springer, New York, 1979)
41. R. Tan, G. Xing, B. Liu, J. Wang, X. Jia, Exploiting data fusion to improve the coverage of wireless sensor networks. IEEE/ACM Trans. Netw. **20**(2), 450–462 (2012)
42. R. Tan, G. Xing, J. Wang, B. Liu, Performance analysis of real-time detection in fusion-based sensor networks. IEEE Trans. Parallel Distrib. Syst. **22**(9), 1564–1577 (2011)
43. C. Taylor, A. Rahimi, J. Bachrach, H. Shrobe, A. Grue, Simultaneous localization, calibration, and tracking in an ad hoc sensor network, in *The 5th International Symposium on Information Processing in Sensor Networks (IPSN)*, Nashville, TN, USA, pp. 27–33 (2006)
44. P.K. Varshney, *Distributed Detection and Data Fusion* (Springer, New York, 1996)
45. P. Volgyesi, G. Balogh, A. Nadas, C.B. Nash, A. Ledeczi, Shooter localization and weapon classification with soldier-wearable networked sensors, in *The 5th International Conference on Mobile Systems Applications and Services (MobiSys)*, San Juan, Puerto Rico, pp. 113–126 (2007)
46. P.J. Wan, C.W. Yi, Coverage by randomly deployed wireless sensor networks. IEEE/ACM Trans. Netw. **14**(Special issue on networking and information theory), 2658–2669 (2006)
47. W. Wang, V. Srinivasan, K.C. Chua, Trade-offs between mobility and density for coverage in wireless sensor networks, in *The 13th Annual International Conference on Mobile Computing and Networking (MobiCom)*, Montreal, QC, Canada, pp. 39–50 (2007)
48. W. Wang, V. Srinivasan, K.C. Chua, B. Wang, Energy-efficient coverage for target detection in wireless sensor networks, in *The 6th International Symposium on Information Processing in Sensor Networks (IPSN)*, Cambridge, MA, USA, pp. 313–322 (2007)
49. G. Xing, R. Tan, B. Liu, J. Wang, X. Jia, C. Yi, Data fusion improves the coverage of wireless sensor networks, in *The 15th Annual International Conference on Mobile Computing and Networking (MobiCom)*, pp. 157–168. ACM, Beijing, China (2009)
50. G. Xing, X. Wang, Y. Zhang, C. Lu, R. Pless, C. Gill, Integrated coverage and connectivity configuration for energy conservation in sensor networks. ACM Trans. Sens. Netw. **1**(1), 36–72 (2005)
51. T. Yan, T. He, J.A. Stankovic, Differentiated surveillance for sensor networks, in *The 1st ACM Conference on Embedded Networked Sensor Systems (SenSys)*, Los Angeles, CA, USA, pp. 51–62 (2003)
52. H. Zhang, J. Hou, On deriving the upper bound of α-lifetime for large sensor networks, in *The 5th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, Tokyo, Japan, pp. 121–132 (2004)
53. Y. Zou, K. Chakrabarty, Sensor deployment and target localization based on virtual forces. in *The 22nd IEEE International Conference on Computer Communications (INFOCOM)*, vol. 2, pp. 1293–1303 (2003)

Part III
Tracking, Estimation, and Counting

Chapter 5
Probabilistic Indoor Tracking of Mobile Wireless Nodes Relative to Landmarks

Ioannis Ch. Paschalidis, Keyong Li, Dong Guo and Yingwei Lin

Abstract The profile-based approach is known to be advantageous when it comes to inferring positions of mobile wireless devices in complex indoor environments. The past decade has seen a significant body of work that explores different implementations of this approach, with varying degrees of success. Here, we cast the profile-based approach in a probabilistic framework. Launching from the theoretical basis that this framework provides, we provide a suite of carefully designed methods that make use of sophisticated computations in pursuit of high localization accuracy with low hardware investment and moderate set-up cost. More specifically, we use full distributional information on signal measurements at a set of discrete locations, termed *landmarks*. Positioning of a mobile node is done relative to the resulting landmark graph and the node can be found near a landmark or in the area between two landmarks. Key elements of our approach include profiling the signal measurement distributions over the coverage area using a special interpolation technique; a two-tier statistical positioning scheme that improves efficiency by adding

Research partially supported by the NSF under grants EFRI-0735974, CNS-1239021, and IIS-1237022, by the ARO under grants W911NF-11-1-0227 and W911NF-12-1-0390, and by the ONR under grant N00014-10-1-0952.

I. Ch. Paschalidis (✉)
Department of Electrical & Computer Engineering., Division of Systems Engineering and Center for Information & Systems Engineering, Boston University, 8 St. Mary's Street, Boston, MA 02215, USA
e-mail: yannisp@bu.edu

K. Li · D. Guo · Y. Lin
Center for Information & Systems Engineering, Boston University, Boston, USA
e-mail: likeyong@ieee.org

D. Guo
e-mail: dong.dongguo@gmail.com

Y. Lin
e-mail: yingwei@bu.edu

H. M. Ammari (ed.), *The Art of Wireless Sensor Networks*,
Signals and Communication Technology, DOI: 10.1007/978-3-642-40066-7_5,

movement detection; and joint clusterhead placement optimization for both localization and movement detection. The proposed system is practical and has been implemented using standard wireless sensor network hardware. Experimentally, our system achieved an accuracy equivalent to less than 5 m with a 95 % success probability and less than 3 m with an 87 % success probability. This performance is superior to well-known contemporary systems that use similar low-cost hardware.

1 Introduction

Demand for reliable indoor positioning of mobile wireless devices is rapidly emerging. Using such a functionality, businesses can better manage their equipment and personnel; museums can provide automatically guided tours and enhance visitor experiences; hospitals can track their patients, personnel, and valuable mobile equipment; rescue workers can navigate through a disaster cite more easily; malls can guide shoppers to specific stores they seek; large warehouses can track their fleet of forklifts [1]; and security agencies can strengthen the protection of critical assets such as nuclear and biochemical materials.

In contrast to indoor positioning, technologies for outdoor positioning are relatively mature. These include the GPS technology that is widely used today, but also technologies using the cellular network (see [2, 3]). Translating, however, these successes to indoor environments is far from straightforward. First, the GPS technology is hardly operational indoors due to heavy signal attenuation by the building structures. Moreover, the cellular-based technologies for outdoor use cannot produce satisfactory positioning accuracy when confronted with the rich effects of the indoor environment on the signals. The indoor environment can be very complex, and also dynamic due to, for example, people moving and doors opening and closing. The triangulation or trilateration techniques, on which GPS or cellular technologies are based, can be inaccurate and ineffective under such conditions.

The main objective of this chapter is to describe a new probabilistic approach to indoor localization of mobile wireless devices. As we will see, the technique is quite general and can handle any type of information from wireless signals that is correlated with location; from very basic information related to signal strength and available with almost any hardware, to more sophisticated information such as angle and time of arrival. The approach we propose is built on rigorous decision theory and that rigor allows us to provide performance guarantees and tackle associated problems such as tracking and optimal system deployment.

1.1 Related Work

Given the interest in indoor positioning, a wide range of indoor positioning solutions have been proposed, with varying degrees of success. Want et al. [4] implemented an

infrared-based positioning system (*Active Badge*) for low-accuracy applications. Priyantha et al. [5] proposed an ultrasound-based system (*Cricket*) that achieved high accuracy, but the system requires the installation of a dense network of ultrasound beacons. Acoustic signals are also used by the system proposed in [6] but without the need of infrastructure nodes. In the wireless positioning camp, many types of signal measurements can be useful, ranging from the most basic—*Received Signal Strength Indication (RSSI)*—to more sophisticated ones including signal phase, time-of-arrival (TOA), angle-of-arrival (AOA), and multipath components (MPC) [7, 8].

Among the RSSI-based approaches, the methods in [9–12] compare mean RSSI measurements to a pre-computed signal-strength map. In particular, [9] tested two methods. The first one uses a map of mean signal strength that is profiled offline, while the second assumes a signal propagation model taking into account how many walls are in the path. The former was shown to be superior, achieving an accuracy of 75 % error <5 m and 50 % <3 m. This system and others succeeded in demonstrating the feasibility of meaningful positioning services using wireless sensor networks and injected enthusiasm into the field. Their performance, though, leaves room for improvement. Many other works followed. Castro et al. [13] improved upon [9] by taking the probabilistic nature of the problem into account through the use of Bayesian network techniques. Another probabilistic method appeared in [14] where both probabilities of RSSI measurements and a hidden Markov model of device movements are being estimated. Lorincz et al. [10] proposed a method based on mean RSSI profiling, but used power-level diversity to achieve an accuracy of 80 % $<$ 3 m. However, further improvement of the performance within this framework (e.g., by increasing the density of "reference signatures") seems unlikely. Another related idea that combines information from GPS at some locations to reduce the need of profiling was proposed in [15] and achieved accuracies between 2 and 7 m. Yet another class of systems such as in [16, 17] use stochastic triangulation techniques but rely on a path loss model, thus, introducing a modeling error. In addition to the evidence provided by [9], our earlier related work based on signal strength profiling [18, 19] has also been shown to reduce the mean error distance (by a factor of about 3.5) compared to stochastic triangulation. References to many other systems can be found in [20] and [21].

1.2 Key Contributions

Despite the extensive literature, several fundamental questions remain. On one hand, there have been some efforts to understand the theoretical limit of wireless-signal-based device positioning. For example, [22] obtained a Cramer-Rao bound of wireless-based positioning, but the result was built upon the presumption that positioning is done using distance and angle of arrival measurements, and that these measurements follow Gaussian distributions. A wider range of techniques were considered by [23], but similar assumptions (especially the Gaussian assumption) were

put in place. Such assumptions might be overly simplistic in the the indoor positioning context. On the other hand, most positioning systems proposed to date have shunned away from sophisticated decision rules for the fear of implementation difficulties on mobile devices. To a certain extent, our work suggests that by careful algorithm design that enables distributed processing, it is possible to implement complex decision rules to achieve robust and accurate positioning using existing low-cost hardware.

Due to the rather diverse applications of indoor positioning and the rapid evolution of the hardware capabilities found in wireless devices, the positioning problem should not be considered in just a single context. Instead, it is meaningful to explore a spectrum of techniques that reflect different constraints and trade-offs of hardware investment, computational complexity, set-up cost, and positioning accuracy. In particular, the present work strives to achieve high accuracy with low hardware investment and moderate set-up cost, at the cost of fairly sophisticated computations. More specifically, our approach does not require advanced signal measurements such as angle and time of arrival (AOA and TOA), and the amount of computation needed is kept under a threshold such that contemporary hardware suffices. One can argue that we treat computational power as a constraint and seek to minimize set-up cost and maximize accuracy.

Even though our primary motivating application is to locate/track nodes of a wireless sensor network, other settings can be served by the exact same techniques we develop. In particular, the nodes can correspond to a wireless phone with Wi-Fi access, or some other Wi-Fi device that we want to locate. Many of the practical applications we outlined earlier can easily be implemented in a smartphone "app," thus, locating and tracking the phone in an indoor environment becomes important.

Our contribution goes beyond a successful positioning/tracking system with attractive experimental results. The approach differs from what has been considered in the literature and is based on a set of formal techniques that result in analytical performance guarantees. As we will see, positioning is done relative to a *landmark graph*. The nodes of this graph are a chosen set of landmarks, or places of interest; and the edge defined between any two nodes corresponds to the contiguous geographical area between the landmarks. Which points in the coverage area are defined to be the landmarks depends on the specifics of the application. What is important is that the device's position is resolved either to a node of the landmark graph, if the device is in its vicinity, or to an edge if the device is in the area between two landmarks. The following topics are then covered:

A. We construct appropriate probabilistic descriptors associated with a device's position from a limited amount of RSSI measurements. This process is commonly known as *profiling*. The descriptors that we construct go beyond mean values and variances to record the shapes of the measurement distributions. We also associate a parametrized family of probability density functions (pdfs) to each location, which introduces some analytical challenges, but proved to produce more robust performance. Our experimental results show that such information is of significant value to performance.

By "a limited amount" of measurements, we mainly mean that one can directly measure the RSSI distributions only for a finite number of positions during the profiling phase. Without much loss of generality, we assume that these positions coincide with the landmarks. In order to construct the probabilistic descriptors of a device's position, we adopt a pdf interpolation technique which originated in statistical physics simulations [24]. To the best of our knowledge, this approach is novel for localization problems. Practices that are equivalent to some form of interpolation are not foreign (e.g., [14]), but these have been carried out rather implicitly without formal design and evaluation. Thus, the present chapter makes a contribution in introducing a formal technique to the field, which is also validated by experimental data.

Earlier versions of probabilistic-descriptors have been explored in [18, 19, 25]. However, the descriptors in [25] are single pdfs rather than pdf families. Although [18] used pdf-family descriptors, the way that they were constructed lacks formalism. A related question to the profiling discussion is: "In what length-scale is pdf interpolation meaningful for RSSI signatures?" Clearly, we would like to minimize measurements but interpolating the pdfs of two very distant locations would not make sense. This question is investigated experimentally, and interestingly, the answer confirms the intuition expressed in earlier work.

B. We develop a two-tier device tracking system that relies on RSSI measurements made by a set of *clusterheads* positioned at some of the landmarks. Clusterheads are simply static nodes positioned at these landmarks with the exact same capabilities as other devices, potentially though with line power to accommodate their heavier use. The motivation for a two-tier system is that we would like to exploit the fact that most mobile agents in indoor environments are on the move only occasionally. Fortunately, we found it possible to detect whether a mobile node has moved from its previously known position based on observations from a single clusterhead. We call this tier *movement detection* (the lower tier). When movement of a device has been detected, the upper tier is invoked, which detects the new position of the device using multiple clusterheads. We call this tier *localization*, as it is how localization is commonly construed. The decisions of both tiers are formulated as composite hypothesis testing problems. We develop the requisite theory and characterize the probability of detection error. Movement detection has a lower run-time cost, and in many applications the device being tracked can remain at one location for a long period of time. The two-tier design thus results in significant savings. We also consider and address the problem of optimally placing clusterheads in order to minimize the probability of making incorrect decisions. Similar results have been largely lacking in the literature except for [18, 19, 25], which considered a single tier. The present chapter extends the optimization to jointly considering localization and movement detection, and establishes that this is computationally feasible.

C. We present a working system that demonstrates the practicality of our approach. Our system achieved an accuracy equivalent to 95 % $<$5 m and 87 % $<$3 m, which should be considered of high-quality compared with well-known

contemporary systems. We also examine the accuracy of our formal pdf interpolation and find that interpolations with two end points 9 m apart may replace empirically measured pdfs with very good precision. This is significant for improving the efficiency of profiling and reinforces the findings in [10].

Notation: We use bold lower case letters for vectors, and bold upper case letters for matrices. Our discussions will involve both probability density functions (pdfs) and probability mass functions (pmfs). With a slight abuse of terminology, we will use the term pdf throughout.

Organization: The remainder of this chapter is organized as follows. Section 2 formulates the problem and identifies the key components of our approach, namely:

- profiling,
- sensor placement,
- localization, and
- movement detection.

Profiling is handled in Sect. 3 and the key technique presented is that of interpolating pdfs. The main theoretical underpinnings of our system and the associated algorithms are presented in Sect. 4. More specifically, Sect. 4.1 reviews some basic facts from information theory which are important in the mathematical development of our algorithms. Section 4.2 describes how (binary) localization and movement detection decisions can be taken by a single clusterhead. Section 4.4 describes how multiple clusterheads can collaborate (in a distributed fashion) to arrive at an accurate localization decision. Section 4.3 presents our approach to optimal clusterhead placement. Section 5 contains all our experimental results and Sect. 6 gathers some concluding remarks.

2 Tracking Problem Formulation

An overview of the tracking problem formulation is shown in Fig. 1.

Consider the problem of tracking a wireless sensor network node in a contiguous space $\mathscr{X}$, which typically corresponds to some indoor environment, e.g., a hospital building, a convention center, or a warehouse. As a way of discretizing this space, we consider a given set of *landmarks* and construct what we call a *landmark graph* as follows. The node set of this undirected graph is the set of landmarks $\mathscr{V} = \{V_i \,|i = 1, \ldots, M\}$. A landmark can be a room, a reception area, a cubical, a storage area or an intersection of aisles. We draw an edge between landmarks that are neighbors (in some geographic sense); i.e., the edge set is $\mathscr{E} = \{E_{ij} \,|i = 1, \ldots, M,\ j > i,\ V_j \in \mathscr{N}_i\}$, where $\mathscr{N}_i$ is the set of neighboring landmarks to V_i. In reality, such an edge may represent a section of a corridor or pathway. There are many different ways to formally define a neighborhood and this is left to the user; one approach could be to set a radius and consider all landmarks within that radius of a node to be its neighbors. With a slight abuse of notation, we sometimes also write: (1) $j \in \mathscr{N}_i$ if

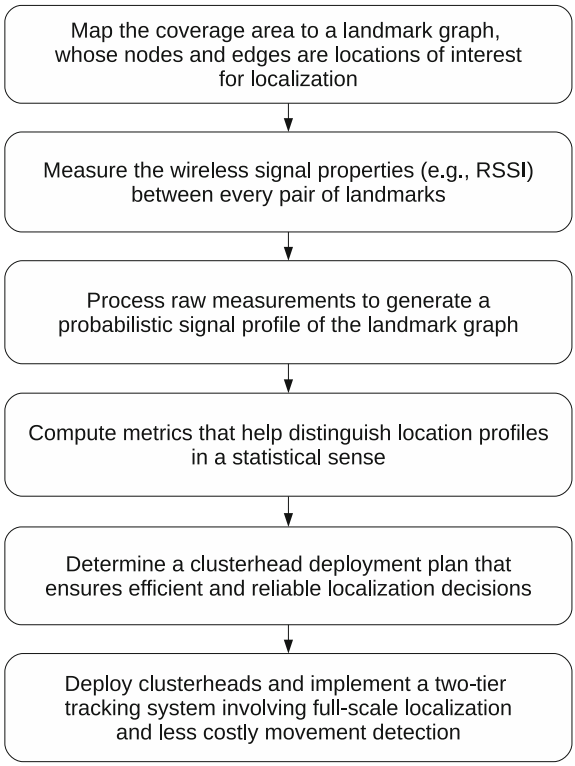


Fig. 1 Overall tracking problem formulation

$V_j \in \mathcal{N}_i$, and (2) $(i, j) \in \mathcal{E}$ when $E_{ij} \in \mathcal{E}$. In what follows, a *location* refers to either a node or an edge. The set of all locations will be denoted by $\mathcal{L} = \{L_l \; | l = 1 \ldots, N\}$, where $N = M + |\mathcal{E}|$.

The next step is profiling, i.e., to associate to various locations appropriate probabilistic descriptors of some features of the wireless signal. Here we use the RSSI, which is measured between all pairs of landmarks. (Additional RF features may also be used if available.) Let $Y^{(k)} \in \{\eta_1, \ldots, \eta_R\}$ be the RSSI received at landmark k, which takes values from an R-dimensional discrete set. We then have a collection of empirical distributions:

$$q_i^{(k)}(y) = \text{Freq}(Y^{(k)} = y|V_i), \qquad y = \eta_1, \ldots, \eta_R, \; i, k = 1, \ldots, M, \tag{1}$$

where k is the index of the receiving landmark and i is the index of the transmitting landmark. In (1), $\text{Freq}(Y^{(k)} = y|V_i)$ simply denotes the fraction of measurements for which $Y^{(k)} = y$. Using these empirical distributions, we build the probabilistic descriptors of all locations using methods introduced in the sections that follow. As the result of profiling, we obtain pdfs of RSSI that characterize the signals transmitted from each location and received at each landmark. In fact, for improved robustness

we associate with each node and edge of the landmark graph a family of pdfs parametrized by vectors $\boldsymbol{\theta}_i$ and $\boldsymbol{\theta}_{ij}$, respectively. These are the location descriptors, or profiles:

$$\begin{aligned} &p_i^{(k)}(\cdot|\boldsymbol{\theta}_i),\ i=1,\ldots,M,\ k=1,\ldots,M; \\ &p_{ij}^{(k)}(\cdot|\boldsymbol{\theta}_{ij}),\ (i,j)\in\mathscr{E},\ k=1,\ldots,M. \end{aligned} \tag{2}$$

We may also list the pdf families in terms of the locations, using the notation

$$p_{Y^{(k)}|\boldsymbol{\theta}_l}(\cdot),\ l=1,\ldots,N,\ k=1,\ldots,M. \tag{3}$$

The former notation will be used when we discuss profiling, while the latter will be used while introducing the decision rules for positioning.

While the wireless signal transmissions are profiled for every pair of landmarks, for the actual operation that follows we do not suggest placing a device at every landmark to listen to the signals transmitted by the mobile nodes; that would be wasteful. Instead, devices for that purpose, which we call "clusterheads," shall be placed only at the "best" $K \leq M$ landmarks where the RSSI measurements carry the greatest amount of location information. The clusterhead placement decisions are naturally based on the results of profiling, and we will cast this problem as a *Mixed Integer Linear Programming (MILP)* problem.

Finally, movement detection and localization are done by "comparing" the clusterheads' RSSI measurements with the location profiles. Formulated as hypothesis testing problems, these decisions are made in statistically meaningful ways.

3 Profiling

This section focuses on how to generate the location profiles, i.e., (2) and (3), using the empirical RSSI distributions, i.e., (1). The key technique is the interpolation of RSSI distributions, or more generally, pdfs.

3.1 Interpolation of Pdfs

A naive way of interpolating pdfs is to calculate a simple weighted average. However, one may quickly find that the naive way can produce unnatural results. For example, given two Gaussian pdfs with different means, their naive interpolation always has two peaks.

A more sophisticated approach has appeared in a work on statistical physics simulation [24], which we adopt with several generalizations. Given K pdfs, $p_1(x), p_2(x), \ldots, p_K(x)$, let $\mu_1, \mu_2, \ldots, \mu_K$ and $\sigma_1^2, \sigma_2^2, \ldots, \sigma_K^2$ be their means and variances, respectively. Let $\boldsymbol{\rho} \in \mathbb{R}^K$ have elements $\rho_1, \rho_2, \ldots, \rho_K \in [0, 1]$

satisfying $\sum_{i=1}^{K} \rho_i = 1$. These are the weights that we assign to the K source pdfs. We are now seeking an interpolated pdf $p_{\boldsymbol{\rho}}(x)$, whose mean and variance are

$$\mu_{\boldsymbol{\rho}} = \sum_{i=1}^{K} \rho_i \mu_i \quad \text{and} \quad \sigma_{\boldsymbol{\rho}}^2 = \sum_{i=1}^{K} \rho_i {\sigma_i}^2. \tag{4}$$

Let

$$\xi_i(x) = \frac{\sigma_i}{\sigma_{\boldsymbol{\rho}}}(x - \mu_{\boldsymbol{\rho}}) + \mu_i, \ i = 1, \ldots, K.$$

Then define

$$p_{\boldsymbol{\rho}}(x) \triangleq \sum_{i=1}^{K} \rho_i \frac{\sigma_i}{\sigma_{\boldsymbol{\rho}}} p_i(\xi_i(x)). \tag{5}$$

Intuitively, this operation can be described as: take a copy of each source pdf, stretch and shift it such that the variance and mean equals $\sigma_{\boldsymbol{\rho}}^2$ and $\mu_{\boldsymbol{\rho}}$ respectively, and finally sum these copies together with the weights adjusted in proportion to the standard deviations of the source pdfs.

We next prove that the mean and variance of $p_{\boldsymbol{\rho}}(x)$ are indeed given by (4). Our proof is different from the one given in [24], but is inspired by the discussions in that paper. We believe our proof is more complete, concise, and intuitive for the present context.

Proposition 3.1 *The pdf given in (5) satisfies*

$$\begin{aligned} &(a) \int p_{\boldsymbol{\rho}}(x)dx = 1, \\ &(b) \int x p_{\boldsymbol{\rho}}(x)dx = \mu_{\boldsymbol{\rho}}, \\ &(c) \int x^2 p_{\boldsymbol{\rho}}(x)dx = \sigma_{\boldsymbol{\rho}}^2 + \mu_{\boldsymbol{\rho}}^2. \end{aligned} \tag{6}$$

Proof: In general, the α-th moment of the pdf in (5) can be written as

$$\begin{aligned} \int x^{\alpha} p_{\boldsymbol{\rho}}(x)dx &= \sum_{i=1}^{K} \rho_i \int x^{\alpha} \frac{\sigma_i}{\sigma_{\boldsymbol{\rho}}} p_i(\xi_i(x))dx \\ &= \sum_{i=1}^{K} \rho_i \int \left[(\xi_i - \mu_i)\frac{\sigma_{\boldsymbol{\rho}}}{\sigma_i} + \mu_{\boldsymbol{\rho}} \right]^{\alpha} p_i(\xi_i)d\xi_i. \end{aligned}$$

The last equality holds because $d\xi_i/dx = \sigma_i/\sigma_{\boldsymbol{\rho}}$. Denote the integral in the above by

$$I_i(\alpha) \triangleq \int \left[(\xi_i - \mu_i)\frac{\sigma_{\boldsymbol{\rho}}}{\sigma_i} + \mu_{\boldsymbol{\rho}} \right]^{\alpha} p_i(\xi_i)d\xi_i.$$

We will show that $I_i(\alpha)$ equals 1, $\mu_{\boldsymbol{\rho}}$, and $\sigma_{\boldsymbol{\rho}}^2 + \mu_{\boldsymbol{\rho}}^2$, respectively for $\alpha = 0$, 1, and 2. Note that if these were true, then $I_i(\alpha)$ is constant with respect to i, thus the weighted sum of $I_i(\alpha)$ over i simply equals $I_i(\alpha)$, and the proposition is proved.

The case of $\alpha = 0$ is trivial.

For $\alpha = 1$,

$$\begin{aligned} I_i(1) &= \int \left[(\xi_i - \mu_i)\frac{\sigma_\rho}{\sigma_i} + \mu_\rho \right] p_i(\xi_i) d\xi_i \\ &= \left[\int \xi_i \, p_i(\xi_i) d\xi_i - \mu_i \right] \frac{\sigma_\rho}{\sigma_i} + \mu_\rho \\ &= (\mu_i - \mu_i)\frac{\sigma_\rho}{\sigma_i} + \mu_\rho \\ &= \mu_\rho. \end{aligned}$$

For $\alpha = 2$,

$$\begin{aligned} I_i(2) &= \int \left[(\xi_i - \mu_i)\frac{\sigma_\rho}{\sigma_i} + \mu_\rho \right]^2 p_i(\xi_i) d\xi_i \\ &= \int \left[(\xi_i - \mu_i)^2 \frac{\sigma_\rho^2}{\sigma_i^2} + 2(\xi_i - \mu_i)\frac{\sigma_\rho}{\sigma_i}\mu_\rho + \mu_\rho^2 \right] p_i(\xi_i) d\xi_i \\ &= \sigma_i^2 \frac{\sigma_\rho^2}{\sigma_i^2} + 2(\mu_i - \mu_i)\frac{\sigma_\rho}{\sigma_i}\mu_\rho + \mu_\rho^2 \\ &= \sigma_\rho^2 + \mu_\rho^2. \end{aligned}$$

In addition to the nice feature of interpolating the mean and variance, formula (5) also interpolates the shapes of the pdfs in some sense. For example, when the original pdfs are Gaussian, the interpolation is also Gaussian.

When the random variable takes discrete values, an issue is that the transformation $\xi(x)$ may produce a value for which probability mass is not defined. An approximate formula that solves this issue is also provided by [24] for the case of interpolating two pdfs. We generalize the method to more than two pdfs. Assume that the probabilities are defined for values $-\infty, \ldots, -1, 0, 1, \ldots, \infty$. For integers j and l, and for $i = 1, \ldots, K$, let

$$\gamma_{ijl} = \max \left\{ 0, \min \begin{bmatrix} \xi_i(j+0.5) \\ l+0.5 \end{bmatrix} - \max \begin{bmatrix} \xi_i(j-0.5) \\ l-0.5 \end{bmatrix} \right\}, \tag{7}$$

where σ_ρ and ξ_i are defined as before. To understand the quantity γ_{ijl}, let $C_{ij,\text{tran}}$ denote the transformed cell $[\xi_i(j-0.5),\ \xi_i(j+0.5)]$, and $C_{l,\text{pmf}}$ denote $[l-0.5,\ l+0.5]$, the cell for which the probabilities are defined. Then γ_{ijl} is the proportion of $C_{l,\text{pmf}}$ that overlaps with $C_{ij,\text{tran}}$; see Fig. 2. Note that the length of $C_{l,\text{pmf}}$ is taken to be 1; otherwise, γ_{ijl} should be normalized by the length of $C_{l,\text{pmf}}$. The approximate formula is then

$$p_\rho(j) = \textstyle\sum_{i=1}^K \rho_i \sum_l \gamma_{ijl} \cdot p_i(l). \tag{8}$$

Formula (7) is chosen such that

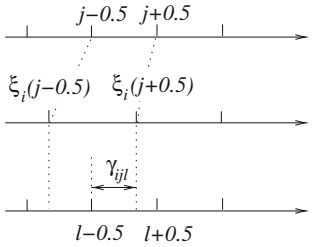


Fig. 2 Interpretation of γ_{ijl}

$$\sum_l \gamma_{ijl} = \frac{\sigma_i}{\sigma_{\boldsymbol{\rho}}},\ \forall i, j,\ \text{and}\ \sum_j \gamma_{ijl} = 1,\ \forall i, l.$$

Formula (8), to which we refer as *linear interpolation*, is what we use in our experiments. Hereafter, we denote the linear interpolation of K pdfs with the coefficient vector $\boldsymbol{\rho} \in \mathbb{R}^K$ by Interpol$(\boldsymbol{\rho}, p_1, p_2, \ldots, p_K)$.

3.2 Associating Pdf Families to Locations

It suffices to consider the RSSI profile of all locations observed by a clusterhead placed at one of the landmarks. The index of the clusterhead is thus suppressed in all formulae of this subsection.

First, we "regularize" the empirical pdfs to eliminate zero elements. This is necessary because the number of our sample measurements during profiling is finite. As a result, some RSSI value η_r that is possible but rare for a location L_i might not be observed during profiling, leaving the rth element of the empirical pdf equal to zero. If we use the empirical pdf directly as the probabilistic descriptor of the location, then when η_r appears, we would rule out L_i immediately, regardless of how many total observations are made and how the rest of the observations resemble the profile of location L_i. This is clearly undesirable. To mitigate this problem, we mix the empirical pdf with a discretized Gaussian-like pdf of the same mean and variance. Namely, let q be an empirical pdf with mean μ and variance σ^2. Let $\phi(\mu, \sigma^2)$ be a Gaussian-like pdf whose domain is discretized to the set $\{\eta_1, \ldots, \eta_R\}$. Let $\gamma \in (0, 1)$ be a chosen mixing factor—typically we set γ to a small value such as 0.1 or 0.2. Then the pdf after regularization is

$$\tilde{q} = \text{Interpol}\left(\begin{bmatrix} 1-\gamma \\ \gamma \end{bmatrix}, q, \phi(\mu, \sigma^2)\right).$$

Second, consider the areas represented by the landmarks. Here, we use the pdf-family framework to achieve robustness. Specifically, suppose V_i has I neighbors—$\mathcal{N}_i = \{V_{j_1}, \ldots, V_{j_I}\}$. Let $\boldsymbol{\theta}_i = (\theta_i^{(1)}, \ldots, \theta_i^{(I)}) \in \mathbb{R}^I$, $\boldsymbol{\theta}_i \geq \mathbf{0}$ elementwise, and $\sum_{j=1}^{I} \theta_i^{(j)} < 1$. Let

$$\boldsymbol{\rho}_{\boldsymbol{\theta}_i} = \begin{pmatrix} 1 - \sum_{j=1}^{I} \theta_i^{(j)} \\ \theta_i^{(1)} \\ \vdots \\ \theta_i^{(I)} \end{pmatrix}.$$

Then the pdf family associated with V_i can be defined as an interpolation of $I + 1$ empirical pdfs:

$$p_i(\cdot|\boldsymbol{\theta}_i) \stackrel{\triangle}{=} \text{Interpol}\left(\boldsymbol{\rho}_{\boldsymbol{\theta}_i}, \tilde{q}_i(\cdot), \tilde{q}_{j_1}(\cdot), \ldots, \tilde{q}_{j_I}(\cdot)\right).$$

This is motivated by the fact than when localization decisions are made, the mobile node we seek to locate will never be exactly on top of a landmark and the pdf of the RSSI measurements may not match well with the pdf associated with that landmark. By interpolating among all pdfs corresponding to neighboring landmarks, we create a pdf family which is very likely to have a member that matches well our measurements.

Last, consider the edges of the landmark graph. For the same reasons we outlined above, we associate with the edge (i, j) another pdf family defined as a set of parameterized interpolations of two pdfs, where the two source pdfs are drawn from the pdf families representing landmarks i and j. Specifically, let $\vartheta_{ij} \in (0, 1)$ and $\boldsymbol{\theta}_{ij}$ be a vector concatenating $\boldsymbol{\theta}_i$, $\boldsymbol{\theta}_j$, and ϑ_{ij}. The pdf family associated with edge (i, j) is

$$p_{ij}(\cdot|\boldsymbol{\theta}_{ij}) \stackrel{\triangle}{=} \text{Interpol}\left(\begin{bmatrix} \vartheta_{ij} \\ 1 - \vartheta_{ij} \end{bmatrix}, p_i(\cdot|\boldsymbol{\theta}_i), p_j(\cdot|\boldsymbol{\theta}_j)\right).$$

The collection of $p_i(\cdot|\boldsymbol{\theta}_i)$'s and $p_{ij}(\cdot|\boldsymbol{\theta}_{ij})$'s corresponds to the pdfs defined in (2), which are then rearranged as in (3).

3.3 An Alternative Interpolation Technique

One may propose another theoretically attractive interpolation technique based on geodesics in the space of probability distributions. We only discuss the two-pdf case for simplicity. A well-known geodesic that connects two pdfs $p_0(x)$ and $p_1(x)$ is a $\rho-$parameterized curve

$$p_\rho(x) = \frac{p_0^{1-\rho}(x) p_1^{\rho}(x)}{\int p_0^{1-\rho}(x) p_1^{\rho}(x) dx}, \tag{9}$$

where $\rho \in [0, 1]$. We will refer to this as the *geometric interpolation*. Like in the linear interpolation case, one can verify that the geometric interpolation of two Gaussian

pdfs remains Gaussian. However, our experiments show that the performance of this interpolation is inferior to that of linear interpolation. The reason is that it emphasizes too much the common support of both source pdfs.

3.4 An Alternative Gaussian Model

In the above, we associate a family of generally shaped pdfs to each location. If a Gaussian model of the RSSI is used instead, this task can be greatly simplified. One may then ask whether using generally shaped pdfs is worth the effort. The answer to this question may depend on circumstances. However, our experiments show that significant information regarding the signals transmitted from a location is captured by our approach, but would be neglected if we assume the Gaussian model. This will be discussed at length later.

Now, having the location profiles in the forms of pdf families, we are in a good position to describe our tracking system.

4 Two-Tier Tracking System

In positioning a mobile wireless node, one typically needs to draw on measurements from multiple (likely more than 3) clusterheads. However, a somewhat surprising observation is: if we already know the previous location of a wireless device and ask the question of whether it has moved to another location, the RSSI signature observed by a *single* clusterhead provides sufficient information. Furthermore, that clusterhead can be selected for each location based on the profiling results. We will refer to such a test that determines whether a device has moved as *movement detection*. Thus, if many mobile nodes in the system actually stay at some locations for prolonged periods of time (e.g., as is the case for office workers), then switching into the movement detection mode can significantly reduce the workload of the system. This motivates a two-tier positioning scheme of the following generic form with parameters T_L, H, and T_M.

Localization Tier: As will be explained, we will make sure that for every pair of locations, there is at least one clusterhead that can distinguish them. It ensures that our system is capable of statistically localizing a mobile device at any location without ambiguity. When a mobile device enters the coverage area, we have an initial set a candidate locations based on the set of clusterheads that can detect the signal from this device. Suppose there are n locations. There is no need to conduct all $n \times (n-1)$ hypothesis tests of these pairs. Instead, we carry out the process in a greedy, single-elimination manner that guarantees the most reliable $n-1$ decision are employed given any deployment. This process is repeated every T_L seconds.

If the device's position remains the same for H rounds of localization, then switch to the movement detection tier.

Movement Detection Tier: Every T_M seconds, use one clusterhead (selected during the initial setup for the mobile device's last known position) to test the hypothesis that the device's position is unchanged. If the hypothesis is rejected, then return to the localization tier. It is worth noting that the single clusterhead, denoted by C, responsible for detecting whether the device remains at a landmark, denoted by L, is selected such that all neighboring landmarks of L can be distinguished from L by the signal profile viewed from C. Most likely, C is at a location that is neither too far nor too close to L.

Next we address the issues of how to make the localization and movement detection decisions and how to ensure that the deployed clusterheads can perform the tasks of both tiers satisfactorily for every location. We will present a generalized framework of statistical decisions that covers both tiers. Furthermore, we optimize both tiers jointly in the clusterhead placement phase.

4.1 Preliminaries

To present the decision rules of the two tiers in a unified framework, we begin by recalling the Kullback-Leibler (KL) distance [26]. For two distribution functions (assuming discrete random variables) $\mathbf{p}$ and $\mathbf{q}$, the KL distance of $\mathbf{q}$ from $\mathbf{p}$ is defined as

$$D(\mathbf{q}\|\mathbf{p}) \triangleq \sum_y q(y) \log \frac{q(y)}{p(y)} \quad (\text{assuming } 0 \cdot \log \frac{0}{a} \equiv 0, \forall a \geq 0). \tag{10}$$

Intuitively, the KL distance reflects the difference between two distributions in a statistically meaningful way (see [27]). It is not a true distance metric (lacks symmetry and does not satisfy the triangle inequality) but it is nonnegative, where zero is achieved if and only if $\mathbf{p} = \mathbf{q}$. Typically, $\mathbf{q}$ is some sample distribution and $\mathbf{p}$ is a model distribution. For example, if a certain event is impossible according to the model distribution ($\mathbf{p}$ has a zero element somewhere) but the event occurred in the sample (the corresponding element of $\mathbf{q}$ is positive), then the KL distance equals ∞, and we immediately know that the sample is not drawn from that model distribution. This example also illustrates that sometimes we may want to avoid proposing a model distribution with a zero element, which is why we regularize the empirical pdfs in Sect. 3.2.

Another important quantity that we draw from information theory is the entropy of the distribution $\mathbf{p}$. Assume a discrete random variable taking values in $\{\eta_1, \ldots, \eta_R\}$. The entropy of the distribution is defined as:

$$H(\mathbf{p}) = -\sum_y p(y) \log p(y). \tag{11}$$

Consider an i.i.d. sequence $\mathbf{y}^n = (y_1, \ldots, y_n)$ drawn from $\mathbf{p}$ and let $\mathbf{q}_{\mathbf{y}^n}$ be its sample (empirical) distribution defined as in (1). It is easy to verify that the probability of $\mathbf{y}^n$ can be written as

$$p(\mathbf{y}^n) = \prod_{i=1}^{R} p(\eta_i)^{nq_{\mathbf{y}^n}(\eta_i)} = e^{-n[H(\mathbf{q}_{\mathbf{y}^n}) + D(\mathbf{q}_{\mathbf{y}^n} \| \mathbf{p})]}. \tag{12}$$

Recall now from Eq. (3) that the pdf family associated with the RSSI from location l to clusterhead k is $p_{Y^{(k)}|\boldsymbol{\theta}_l}(\cdot),\ l = 1, \ldots, N,\ k = 1, \ldots, M$, as constructed in the profiling phase. For either localization or movement detection, suppose the clusterhead makes n i.i.d. observations $\mathbf{y}^{(k),n} = (y_1^{(k)}, \ldots, y_n^{(k)})$, with a corresponding sample distribution $\mathbf{q}_{\mathbf{y}^n}^{(k)}$, upon which the decision rules apply. The efficiency of the decision rules to be presented will be quantified using the *error exponent*, defined as

$$d \stackrel{\triangle}{=} -\lim_{n\to\infty} \sup \frac{1}{n} \log \mathbf{P}(\text{error}). \tag{13}$$

This represents the exponentially decay rate at which the probability of error converges to zero.

4.2 Statistical Decisions Contributed by a Single Clusterhead

Next, consider the decisions that can be made by one clusterhead (hence the clusterhead index is dropped). For localization, the core task is to distinguish between two candidate locations of the mobile device. It was shown in [18, 19] that the question boils down to a binary composite hypothesis testing problem (composite because the distributions involved have unknown parameters—the $\boldsymbol{\theta}$'s), and one can use the well-known *Generalized Likelihood Ratio Test (GLRT)*. In particular, the GLRT decides location i over j if

$$\frac{1}{n} \log \frac{\max_{\boldsymbol{\theta}_j} p_{Y|\boldsymbol{\theta}_j}(\mathbf{y}^n)}{\max_{\boldsymbol{\theta}_i} p_{Y|\boldsymbol{\theta}_i}(\mathbf{y}^n)} < \lambda_{ij},$$

for some appropriate threshold λ_{ij}. We have

$$\begin{aligned}
\lambda_{ij} &> \frac{1}{n} \log \max_{\boldsymbol{\theta}_j} p_{Y|\boldsymbol{\theta}_j}(\mathbf{y}^n) - \frac{1}{n} \log \max_{\boldsymbol{\theta}_i} p_{Y|\boldsymbol{\theta}_j}(\mathbf{y}^n) \\
&= \max_{\boldsymbol{\theta}_j} [-H(\mathbf{q}_{\mathbf{y}^n}) - D(\mathbf{q}_{\mathbf{y}^n} \| p_{Y|\boldsymbol{\theta}_j})] \\
&\quad - \max_{\boldsymbol{\theta}_i} [-H(\mathbf{q}_{\mathbf{y}^n}) - D(\mathbf{q}_{\mathbf{y}^n} \| p_{Y|\boldsymbol{\theta}_i})] \\
&= \min_{\boldsymbol{\theta}_i} D(\mathbf{q}_{\mathbf{y}^n} \| p_{Y|\boldsymbol{\theta}_i}) - \min_{\boldsymbol{\theta}_j} D(\mathbf{q}_{\mathbf{y}^n} \| p_{Y|\boldsymbol{\theta}_j}).
\end{aligned}$$

Thus, in terms of the KL distance, the GLRT rule can be expressed as:

$$\begin{aligned} &\text{Decide location } i \text{ relative to } j \text{ if and only if} \\ &\quad \min_{\boldsymbol{\theta}_i} D(\mathbf{q}_{\mathbf{y}^n} \| p_{Y|\boldsymbol{\theta}_i}) - \min_{\boldsymbol{\theta}_j} D(\mathbf{q}_{\mathbf{y}^n} \| p_{Y|\boldsymbol{\theta}_j}) < \lambda_{ij}. \end{aligned} \tag{14}$$

A slightly conservative estimate of the corresponding error exponent, denoted by d_{ij}, was also derived in [19] (including an explanation of why λ_{ij} is not always zero and how it can be selected). Moreover, [19] describes how a decision among many potential locations can be taken by using a sequence of binary tests; interestingly enough, such a decision can be taken in a distributed manner by appropriate collaboration between the clusterheads.

Let us now turn our attention to the movement detection tier. Suppose that the last known location of a mobile device is location j and we would like to determine if the device remains at j based on the n i.i.d. observations in $\mathbf{y}^n$ by a single clusterhead. The following *Generalized Hoeffding Test (GHT)* [28], expressed analogously to (14) using the KL distance, is applicable:

$$\begin{aligned} &\text{Report ``no movement'' if and only if} \\ &\quad \min_{\boldsymbol{\theta}_j} D(\mathbf{q}_{\mathbf{y}^n} \| p_{Y|\boldsymbol{\theta}_j}) < \lambda_j. \end{aligned} \tag{15}$$

Let α_j and β_j be the error probabilities of false alarm and missed detection, respectively. The following proposition shows that the GHT is optimal in a *Generalized Neyman-Pearson* sense.

Lemma 4.1 *The test in (15) is optimal in a generalized Neyman-Pearson sense, that is,*

$$\limsup_{n\to\infty} \frac{1}{n} \log \alpha_j < -\lambda_j, \qquad \forall \boldsymbol{\theta}_j, \tag{16}$$

and $-\limsup_{n\to\infty} \frac{1}{n} \log \beta_j$ *is maximized among all tests satisfying (16) uniformly for all* $\boldsymbol{\theta}_i$ *characterizing some alternative pdf.*

Proof: Let $\mathbf{P}_j(\cdot)$ denote a probability conditional on the mobile node having not moved from location j. Let also $\mathcal{Q}_n = \{\boldsymbol{\nu} \mid \boldsymbol{\nu} = \mathbf{q}_{\mathbf{y}^n} \text{ for some } \mathbf{y}^n\}$ denote the set of all empirical measures that can be obtained from an n-length observation sequence and $T_n(\boldsymbol{\nu}) = \{\mathbf{y}^n \mid \mathbf{q}_{\mathbf{y}^n} = \boldsymbol{\nu}\}$ the set of n-length observation sequences with an empirical measure equal to $\boldsymbol{\nu}$.

First, letting $\mathbf{P}_{\boldsymbol{\nu}}$ denote a probability under the measure $\boldsymbol{\nu}$ and $|\cdot|$ the cardinality, wehave

$$\begin{aligned} \mathbf{P}_{\boldsymbol{\nu}}[T_n(\boldsymbol{\nu})] &= \sum_{\{\mathbf{y}^n | \mathbf{q}_{\mathbf{y}^n} = \boldsymbol{\nu}\}} \mathbf{P}_{\boldsymbol{\nu}}[\mathbf{y}^n] \\ &= \sum_{\{\mathbf{y}^n | \mathbf{q}_{\mathbf{y}^n} = \boldsymbol{\nu}\}} \nu(\eta_1)^{n\nu(\eta_1)} \cdots \nu(\eta_R)^{n\nu(\eta_R)} \end{aligned}$$

$$= |T_n(\boldsymbol{\nu})| e^{-nH(\boldsymbol{\nu})},$$

which implies

$$|T_n(\boldsymbol{\nu})| \leq e^{nH(\boldsymbol{\nu})}. \tag{17}$$

Now, for all $\boldsymbol{\theta}_j$ the false alarm probability is given by

$$\begin{aligned}
\alpha_j &= \mathbf{P}_j[\{\mathbf{y}^n \mid \min_{\boldsymbol{\theta}_j} D(\mathbf{q}_{\mathbf{y}^n} \| p_{Y|\boldsymbol{\theta}_j}) \geq \lambda_j\}] \\
&= \sum_{\{\mathbf{q}_{\mathbf{y}^n} | \min_{\boldsymbol{\theta}_j} D(\mathbf{q}_{\mathbf{y}^n} \| p_{Y|\boldsymbol{\theta}_j}) \geq \lambda_j\}} |T_n(\mathbf{q}_{\mathbf{y}^n})| \; p_{Y|\boldsymbol{\theta}_j}(\mathbf{y}^n) \\
&\leq \sum_{\{\mathbf{q}_{\mathbf{y}^n} | \min_{\boldsymbol{\theta}_j} D(\mathbf{q}_{\mathbf{y}^n} \| p_{Y|\boldsymbol{\theta}_j}) \geq \lambda_j\}} e^{nH(\mathbf{q}_{\mathbf{y}^n})} e^{-n[H(\mathbf{q}_{\mathbf{y}^n}) + D(\mathbf{q}_{\mathbf{y}^n} \| p_{Y|\boldsymbol{\theta}_j})]} \\
&= \sum_{\{\mathbf{q}_{\mathbf{y}^n} | \min_{\boldsymbol{\theta}_j} D(\mathbf{q}_{\mathbf{y}^n} \| p_{Y|\boldsymbol{\theta}_j}) \geq \lambda_j\}} e^{-nD(\mathbf{q}_{\mathbf{y}^n} \| p_{Y|\boldsymbol{\theta}_j})} \\
&\leq (n+1)^R e^{-n\lambda_j}.
\end{aligned}$$

For the first inequality above we have used (12) and (17). In the last inequality above we used the fact that the set of all possible empirical measures, $\mathscr{Q}_n$, has cardinality upper bounded by $(n+1)^R$ (a symbol of length R with each element taking values from $\{\frac{0}{n}, \ldots, \frac{n}{n}\}$). This establishes (16).

Let now $\mathscr{S}_n$ be some other decision rule satisfying (16). It is well known that the empirical measure is a sufficient statistic so any rule will depend only on that. Let $\alpha_{\mathscr{S}_n}$ and $\beta_{\mathscr{S}_n}$ denote the corresponding false alarm and missed detection probabilities. For all $\epsilon > 0$ and all large enough n we have

$$\alpha_{\mathscr{S}_n} \leq e^{-n(\lambda_j + \epsilon)}. \tag{18}$$

Meanwhile for all $\epsilon > 0$, all large enough n, and any $\mathbf{y}^n$ such that $\mathscr{S}_n$ declares "movement" it holds

$$\begin{aligned}
\alpha_{\mathscr{S}_n} &= \sum_{\{\mathbf{q}_{\mathbf{y}^n} | T_n(\mathbf{q}_{\mathbf{y}^n}) \text{ implies movement}\}} |T_n(\mathbf{q}_{\mathbf{y}^n})| \; p_{Y|\boldsymbol{\theta}_j}(\mathbf{y}^n) \\
&\geq \sum_{\{\mathbf{q}_{\mathbf{y}^n} | T_n(\mathbf{q}_{\mathbf{y}^n}) \text{ implies movement}\}} (n+1)^{-R} e^{-nD(\mathbf{q}_{\mathbf{y}^n} \| p_{Y|\boldsymbol{\theta}_j})} \\
&\geq e^{-n[D(\mathbf{q}_{\mathbf{y}^n} \| p_{Y|\boldsymbol{\theta}_j}) + \epsilon]},
\end{aligned}$$

where the first inequality above uses [29, Lemma 2.1.8]. Comparing the above with (18) it follows that if $\mathbf{y}^n$ implies "movement" then for all $\boldsymbol{\theta}_j$ it should hold $D(\mathbf{q}_{\mathbf{y}^n} \| p_{Y|\boldsymbol{\theta}_j}) \geq \lambda_j$. Therefore, the GHT in (15) should declare movement as well, which implies that $\beta_{\mathscr{S}_n} \geq \beta_j$ for all $\boldsymbol{\theta}_i$ characterizing an alternative pdf, where

$i \neq j$. The latter establishes that the GHT maximizes the exponent of the missed detection probability.

Eq. (16) provides a bound on the exponent of the false alarm probability. To bound the exponent of the missed detection probability we can use Sanov's theorem [29, Chap. 2]. Specifically,

$$\limsup_{n\to\infty} \frac{1}{n} \log \beta_j \leq - \min_{\mathbf{q}\in\mathscr{H}_j} D(\mathbf{q}\|p_{Y|\boldsymbol{\theta}_i}), \ \forall \boldsymbol{\theta}_i, \tag{19}$$

where $\mathscr{H}_j = \{\mathbf{q}| \min_{\boldsymbol{\theta}_j} D(\mathbf{q}\|p_{Y|\boldsymbol{\theta}_j}) < \lambda_j\}$.

Defining

$$Z_{j,\boldsymbol{\theta}_i}(\lambda_j) = \begin{array}{ll} \min_{\mathbf{q}} & D(\mathbf{q}|p_{Y|\boldsymbol{\theta}_i}) \\ \text{s.t.} & \min_{\boldsymbol{\theta}_j} D(\mathbf{q}\|p_{Y|\boldsymbol{\theta}_j}) < \lambda_j, \end{array} \tag{20}$$

and

$$Z_j(\lambda_j) = \min_{i\neq j} \min_{\boldsymbol{\theta}_i} Z_{j,\boldsymbol{\theta}_i}(\lambda_j), \tag{21}$$

we can write

$$-\limsup_{n\to\infty} \frac{1}{n} \log \beta_j \geq Z_j(\lambda_j).$$

The error exponent of the whole test equals the lesser between the exponents of α_j and β_j. On the other hand, it is straightforward to show that $Z_j(\lambda_j)$ is monotonically decreasing in λ_j. If we could compute $Z_j(\lambda_j)$, then the solution of $Z_j(\lambda_j) = \lambda_j$ gives the optimal value of λ_j as well as the best achievable error exponent. However, the computational cost of finding the exact solution is significant. This is mostly because the constraint in (20) is non-convex and also high-dimensional. To side-step this problem, we replace $Z_{j,\boldsymbol{\theta}_i}(\lambda_j)$ with an estimate based on dual relaxation:

$$\tilde{Z}_{j,\boldsymbol{\theta}_i}(\lambda_j) \stackrel{\triangle}{=} \max_{\mu\geq 0} \ \min_{\boldsymbol{\theta}_j} \ \begin{array}{ll} \min_{\mathbf{q}} & D(\mathbf{q}|p_{Y|\boldsymbol{\theta}_i}) + \mu(D(\mathbf{q}\|p_{Y|\boldsymbol{\theta}_j}) - \lambda_j) \\ \text{s.t} & \sum_y q(y) = 1, \\ & q(y) \geq 0, \qquad \forall y. \end{array} \tag{22}$$

As a relaxation, it holds that $\tilde{Z}_{j,\boldsymbol{\theta}_i}(\lambda_j) \leq Z_{j,\boldsymbol{\theta}_i}(\lambda_j)$. The computational gain comes from the fact that the minimization over $\mathbf{q}$ is now convex. In fact, the optimal solution can be found in closed-form:

$$q^*(y) = \frac{p^{\rho}_{Y|\boldsymbol{\theta}_i}(y) p^{1-\rho}_{Y|\boldsymbol{\theta}_j}(y)}{\sum_\eta p^{\rho}_{Y|\boldsymbol{\theta}_i}(\eta) p^{1-\rho}_{Y|\boldsymbol{\theta}_j}(\eta)}, \quad \text{where} \quad \rho = \frac{1}{1+\mu}. \tag{23}$$

Then $\tilde{Z}_j(\lambda_j) = \min_{i \neq j} \min_{\theta_i} \tilde{Z}_{j,\theta_i}(\lambda_j)$, which is also monotonically decreasing in λ_j. The solution of $\tilde{Z}_j(\lambda_j) = \lambda_j$, denoted by h_j, gives a near-optimal threshold and the error exponent at the same time.

4.3 Optimal Clusterhead Placement

From the above, for each landmark (also candidate clusterhead location) k, we have a collection of localization error exponents d_{ijk}, and movement-detection error exponents h_{jk}, all of which are optimized (note that the index of the clusterhead location is now added back).

In the profiling phase, every location is a potential place of clusterhead installation. In the actual operation of the tracking system, however, we may not want to place a clusterhead at every location, because if we do that, some of the clusterheads may not be in a good position to provide useful information. The budget of the system may further reduce the number of clusterheads that we are allowed to deploy. For these reasons, we need to place the clusterheads carefully. In the case of the dense deployment mentioned above, placement amounts to selecting which nodes from the ones already deployed will play the role of a clusterhead.

In formulating the clusterhead placement problem, our goal is to ensure:

1. For the localization tier, there should be at least one clusterhead k for each pair of locations i, j such that L_i, L_j can be clearly distinguished using observations of clusterhead k under the GLRT.
2. For the movement detection tier, there should be at least one clusterhead k for each location j such that L_j can be distinguished from alternative locations using clusterhead k under the GHT.

More precisely, for a given number K of clusterheads to deploy, we maximize the error exponent level ϵ that is met (or exceeded) for every location in the movement detection tier, and for every pair of locations in the localization tier. Furthermore, the error exponent for each location or each pair of locations is given by the clusterhead that can best resolve the case (with greatest error exponent). Formally, let $\mathscr{P}$ denote the set of landmarks where clusterheads are placed. The problem can be written as

$$\begin{aligned} \max_{\mathscr{P} \subset \mathscr{V}} \ & \epsilon \\ \text{s.t. } & |\mathscr{P}| = K, \\ & \min_{i,j} \max_k d_{ijk} \geq \epsilon, \\ & \min_j \max_k h_{jk} \geq \epsilon. \end{aligned} \tag{24}$$

It is important to note that the optimal value of this problem provides an upper bound on the probability of error for both localization and movement detection decisions. That is, if ϵ^* is an optimal solution, the maximum probability of error of our system satisfies

$$
\begin{aligned}
\max \quad & \epsilon \\
\text{s.t.} \quad & \textstyle\sum_{k=1}^{M} x_k = K \\
& \textstyle\sum_{k=1}^{M} y_{ijk} = 1, \ \forall i, j = 1, \ldots, N, i < j, \\
& \textstyle\sum_{k=1}^{M} z_{jk} = 1, \ \forall j = 1, \ldots, N, \\
& y_{ijk} \leq x_k, \ \forall i, j, i < j, k = 1, \ldots, M, \\
& z_{jk} \leq x_k, \ \forall j, k = 1, \ldots, M, \\
& \epsilon \leq \textstyle\sum_{k=1}^{M} d_{ijk} y_{ijk}, \ \forall i, j, i < j, \\
& \epsilon \leq \textstyle\sum_{k=1}^{M} h_{jk} z_{jk}, \ \forall j, \\
& y_{ijk} \geq 0, \ \forall i, j, i < j, \ \forall k, \\
& z_{jk} \geq 0, \ \forall j, \ \forall k, \\
& x_k \in \{0, 1\}, \ \forall k.
\end{aligned}
$$

Fig. 3 Clusterhead placement MILP formulation

$$\limsup_{n \to \infty} \frac{1}{n} \log \mathbf{P}[\text{error}] \leq -\epsilon^*. \tag{25}$$

Problem (24) can be reformulated as the *Mixed Integer Linear Programming (MILP)* problem shown in Fig. 3.

Software exists for solving generic MILPs. Furthermore, we have designed an algorithm that solves the MILP in Fig. 3 much faster than commercial general purpose MILP-solvers. Our approach is to solve this problem by an iterative *feasibility* algorithm along the lines proposed in [30]. In particular, we use a modified version of a *two-phase* algorithm proposed in [31]. Its computational advantage lies in the fact that we solve a feasibility problem in each iteration that contains only $O(M)$ variables and $O(N^2)$ constraints, instead of the $O(N^2M)$ variables and $O(N^2M)$ constraints that appear in the formulation.

4.4 Putting Everything Together: The Decision Procedure Involving Multiple Deployed Clusterheads

While movement-detection requires one clusterhead at a time, localization generally requires more than one clusterhead. No matter how good the decision rule is, a single clusterhead is normally not enough for distinguishing every pair of locations in a wide coverage area, because given any single clusterhead, there may be pairs of locations whose RSSI profiles are similar. So, we use the following procedure to "chain" together the decisions of multiple clusterheads.

When a wireless device is detected, we can have an initial estimate of the candidate locations based on the set of clusterheads that received signals from that device.

Suppose there are n candidate location. Instead of comparing all $n \times (n-1)$ pairs of them, we apply a greedy selection process.

To understand this procedure, first recall that the profiling and clusterhead placement phases assigned one clusterhead for each pair of locations that need to distinguished. Furthermore, the statistical reliability with which this pair of locations can be distinguished (based on the clusterhead of our choosing) is associated with an error exponent, the greater value of which indicates higher reliability.

Now, we sort the pairs of candidates in descending order of the associated error exponent. Using GLRT, we first make a judgment between the first pair of candidates in this ordered list. That is, we start with the most reliable decision that can be made. The candidate that loses the judgment is eliminated, so are the pairs that include the eliminated candidate. Then we move our focus to the remaining pair that is associated with the next greatest error exponent. In the end, $n-1$ judgments are made and a single winner is left.

5 Experiments

Our testbed is set up on the first floor of a Boston University building (see Fig. 4). The wireless device used is the MPR2400 (MICAz) "mote" from Crossbow Technology Inc. (now Memsic Inc.).

5.1 Testing the Complete System

Our two-tier tracking system covers 10 rooms and the corridors, which are mapped to 30 landmarks, marked by either a green circle or a red square on the floor plan (Fig. 4), where the latter marks the clusterhead positions obtained from solving the optimal placement problem. The landmark graph is then constructed resulting in 39 edges. Adding the landmarks and edges together, we have a total of 69 locations.

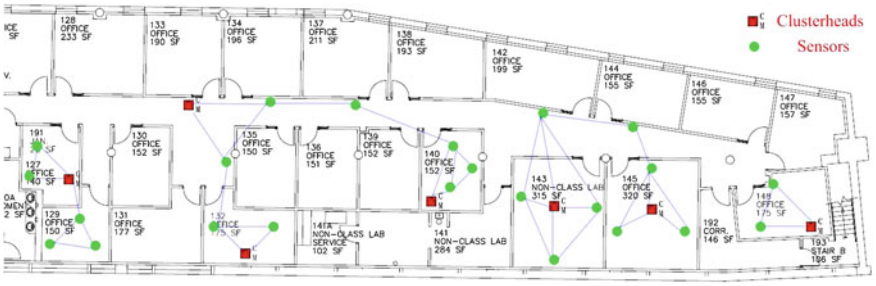


Fig. 4 Floor plan with the landmarks for the testbed

Hence, $N = 69$ and $M = 30$ in this experiment. A mote is placed at each landmark location, but only some of them will serve as clusterheads. All 30 motes are connected to a base MICAz through a mesh network. The base mote is docked on a Stargate node which forwards the messages back to the server.

The experimental validation of our localization approach can be divided into the following six phases, which have a one-to-one correspondence to those in Fig. 1:

Phase 1: Map the coverage area to a landmark graph.

Phase 2: We obtained the empirical pdfs for the landmarks corresponding to Eq. (1). The 30 motes placed at the landmark locations took turns to broadcast packets, specifically, when one was transmitting the others were listening and recording the RSSI. A total of 200 packets were transmitted by each mote. The data collection was repeated for the combinations of two frequencies and two power levels; details will be given below.

Phase 3: We used the methods in Sect. 3 to construct the pdf families corresponding to Eq. (2), which are the descriptors of all 69 locations. Note that the interpolation technique allowed us to construct high quality descriptors without densely covering the area with landmarks.

Phase 4: We obtained d_{ijk} and h_{jk} as described in Sec. 4.

Phase 5: We solved the MILP to optimize clusterhead placement and simultaneously obtained the performance guarantee of Eq. (25). In the MILP formulation, we needed to input K, the total number of clusterheads. By varying K from 1 to 30, we discovered that the performance guarantee reached a satisfactory level after $K = 7$, and somewhat flattens afterward. Thus, we placed clusterheads at 7 locations (again, marked by red squares in Fig. 4).

Phase 6: We introduced mobile motes in the coverage area and let the system make localization and movement detection decisions.

For Phase 1, the coverage area is mapped to a landmark graph as shown in Fig. 4.

We let Phase 2 (a completely automated procedure) stretch over 24 h to acquire data under diverse conditions of the surrounding environment. The objective is to capture the indoor environment in all possible "modes" and configurations so that the pdfs we generate can model any one of these conditions. Phase 3 takes virtually no time. Phase 4 takes another 24 h on our computer, although further optimization of our code may reduce the computation time significantly. Phase 5 only takes about half an hour. Note that all these steps are performed only once, after which the amount of real-time computation needed for each localization decision is very small, such that the resources on the clusterheads (typically motes plugged into a wall outlet or a large enough battery) are sufficient.

We know from previous experiences [19] and the literature that frequency and power diversity provide better performance. The mote to be located broadcasted 20 packets over the combination of 2 frequencies (2.410 GHz and 2.460 GHz) and 2 power levels (0 dBm and -10 dBm), with 5 packets corresponding to each combination. We achieved a mean error distance of 87.32 inches, which is better than our earlier result of 96.08 inches [19] based on techniques that do not use a formal method of pdf interpolation. The percentile of errors < 3 m (118 inches) also

improved from 80 to 87 %. One may also count from Fig. 5 that the percentile of errors $<$ 5 m (197 inches) is 95 %.

The total coverage area (we have excluded the rooms that are in the floor plan but to which we did not have access) was 1827 feet^2, that is, about 61 feet^2 per landmark. With a mean error distance of $\bar{D}_e = 7.3$ feet the mean area of "confusion" was $7.3^2 = 53$ feet^2. It is evident that we were able to achieve accuracy on the same order of magnitude as the area "covered" by a landmark; this is the best possible outcome with a "discretized" system such as ours. That is, the system was identifying the correct location or a neighboring location most of the time. We used a clusterhead density of 1 clusterhead per $1827/7 = 261$ feet^2. Note that our system is *not* based on the "proximity" to a clusterhead; the ratio of locations to clusterheads is $69/7$, or about 10.

We also obtained results for the movement detection tier. The mote to be monitored now transmits 20 packets at a single frequency and power level depending on its *a priori* location. The use of a single frequency/power level was selected because the diversity only benefits decisions involving multiple clusterheads. The resulting error probability was 8 %. Again, we emphasize that we are exploiting only the most basic RF measurements in obtaining these results. Yet, the approach is easily generalizable to include additional information if available (in that case, instead of scalar RSSI measurements we will be dealing with an observation vector).

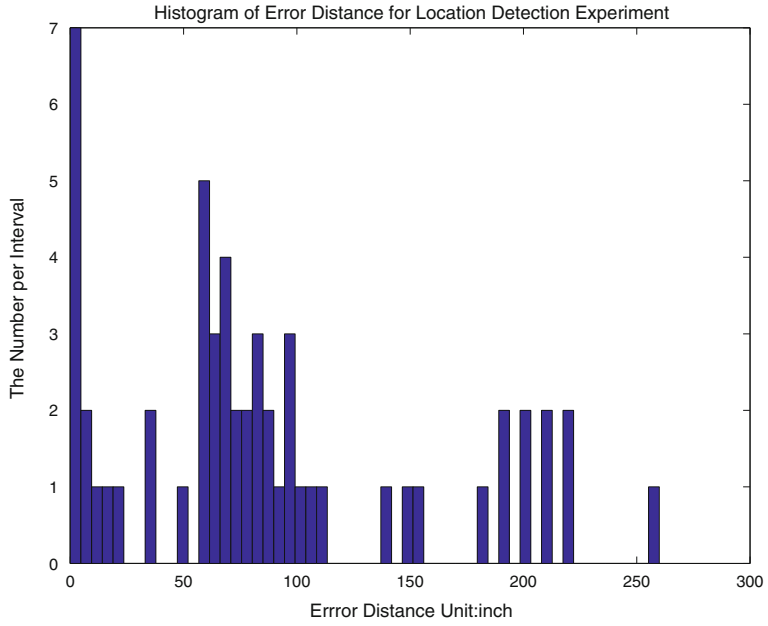


Fig. 5 Localization result

5.2 A Closer Look at Pdf Interpolation

We have proposed a rather sophisticated interpolation technique for generating location profiles. One concern is: if the interpolated pdfs were merely low-quality approximations of the actual pdfs, then we might be better-off using a Gaussian approximation, which is computationally cheaper. In our experiments however, the interpolated pdfs did a very good job preserving the shape information of the empirical pdfs which were not close to a Gaussian. As will be shown, the decision accuracy using the interpolated pdfs dominates that of the Gaussian approximation by a significant margin. Another question that we attempt to answer is: At what length-scale does pdf interpolation make sense? It turns out that the interpolation is very meaningful when the two end points are about 30 feet (or 9 m) apart, but not when they are 60 feet apart.

The first experiment is conducted in a roughly straight corridor of about 60 feet long, mapped to 5 locations roughly 15 feet apart. Labeling the locations consecutively as location $1, 2, \ldots, 5$, we place the clusterhead at location 1 (which is at one end of the corridor). To measure the signals transmitted from each location, one of the coauthors stood at that location holding a transmitting mote, which sends a packet every 5 seconds. We chose to have a person hold the mote because this is close to an actual application scenario. The clusterhead received the packets and recorded the RSSI values. During the experiment, a total of 150 packets were sent from each location. Due to packet loss, the number of actual samples taken by the clusterhead is less, but we still obtained more than 100 samples for each location. Then, we mix a Gaussian component into each of the six empirical distributions as described earlier with a mixing factor of 0.2, i.e., regularized empirical distribution = 0.8 measured + 0.2 Gaussian. The empirical distributions for the six locations after regularization are denoted by $q_1, q_2, \ldots, q_5$.

We compare three interpolation methods. First, in what is labeled "linear short interpolation," the interpolated pdf of location i is generated using q_{i+1} and q_{i-1}:

$$p_{i,\text{short}} = \text{Interpol}\left(\begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}, q_{i-1}, q_{i+1}\right), \ i = 2, 3, 4.$$

Second, in what is labeled "linear long interpolation," the interpolated pdfs are generated using q_1 and q_5:

$$p_{i,\text{long}} = \text{Interpol}\left(\begin{bmatrix} \dfrac{5-i}{4} \\ \dfrac{i-1}{4} \end{bmatrix}, q_1, q_5\right), \quad i = 2, 3, 4.$$

Third, we adopt the Gaussian model instead and interpolate the pdf of each location with adjacent locations:

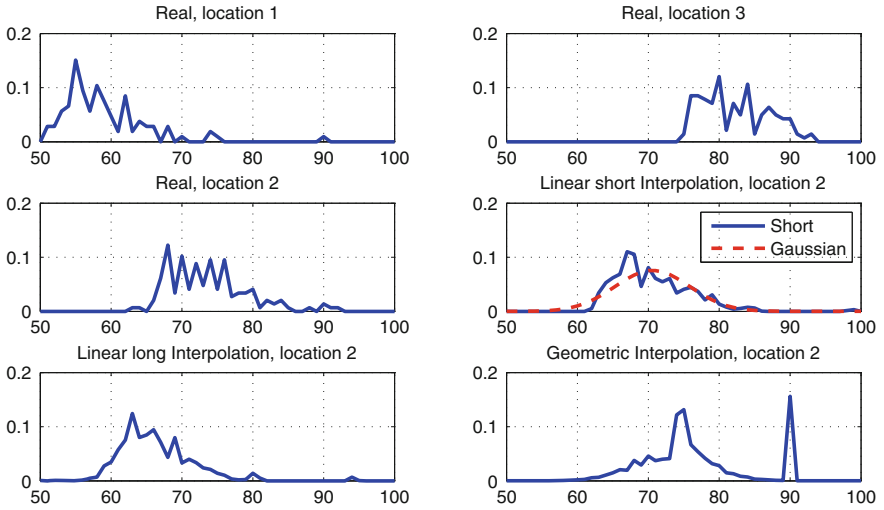


Fig. 6 Visual comparison of interpolated pdfs for location 2. The horizontal axis represents signal strength reading of the motes

$$p_{i,\text{gaussian}} = \text{Interpol}\left(\begin{bmatrix}0.5\\0.5\end{bmatrix}, \phi(\mu_{i-1}, \sigma^2_{i-1}), \phi(\mu_{i+1}, \sigma^2_{i+1})\right)$$
$$= \phi\left(\frac{\mu_{i-1}+\mu_{i+1}}{2}, \frac{\sigma^2_{i-1}+\sigma^2_{i+1}}{2}\right), \qquad i = 2, 3, 4.$$

Last, we also include the geometric interpolation (cf. Sec. 3.3) for comparison, denoted by $p_{i,\text{geo}}$, where location $i \in \{2, 3, 4\}$ is interpolated by location $i-1$ and $i+1$ as in the linear short case.

(1) Qualitative Study: We visually compare the various interpolation results for location 2 as an example (Fig. 6). The short interpolation seems to capture some shape information of the actual pdf that is missed by the Gaussian model. For example, the empirical pdf is skewed to the left. The interpolated pdf also exhibits the skewness, while the Gaussian pdf is always symmetrical. One may also notice that the linear long and the geometric interpolations appear very different from the actual pdf.

(2) Quantitative Observation: First, it is of distance between pdfs. As we have seen, the KL distance appears in both localization and movement detection error exponents.

The comparison is plotted in Fig. 7. It is very interesting to see that *the quality of linear short interpolation dominates that of the Gaussian model in the KL sense*. For example, the KL distance of short-interpolation-to-empirical for location 4 is

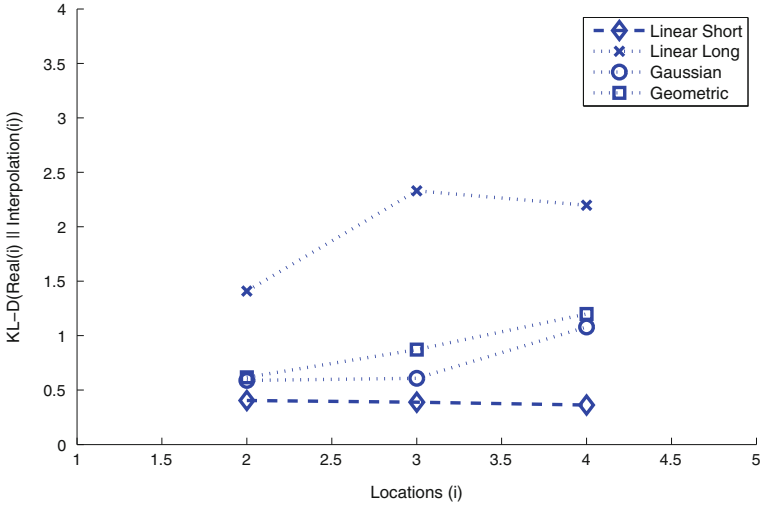


Fig. 7 Kullback-Leibler distance comparison

only a little over one-third of that of the Gaussian model. For locations 2 and 3, the difference is roughly a factor of 1.5, which is still significant. Similarly, the linear short interpolation appears superior to the geometric (also short) interpolation. The long interpolation on the other hand clearly departs from the actual distribution.

Next, we attempted location distinction using these pdfs. We omit the geometric and the linear long cases as, based on the discussion above, they are not suitable for our purposes. We will hence change the label "linear short interpolation" to simply "linear interpolation." For each location $j = 2, 3$, and 4, and each sample size $n = 5, 10, 15, \ldots, 30$, we tested the hypothesis that "the wireless node is at location j" using n RSSI measurements drawn randomly from a large pre-compiled data set. Each sample contained measurements associated with one single location i, where $i \in \{1, \ldots, 6\}$. The GHT was used to make the decision, in which the threshold λ was optimized for each j and n. For each pair of j and n, we repeated the trial 4000 times and calculated the empirical error probability, which is a weighted sum of all types of errors. The result is shown in Fig. 8.

Several observations are in order. First, very low error probability is achieved using the linearly interpolated pdfs. Second, and maybe more interestingly, the error probabilities using linearly interpolated pdfs are lower than those using the Gaussian model in all instances, and the difference is quite large in most cases. Further, as the sample size increases, the former decreases faster than the latter. These demonstrate that the approximation qualities of the interpolated pdfs are fairly notable, and the added computational effort (compared to the Gaussian model) may indeed be well justified.

Yet another interesting observation emerges in the comparison with [10], which has shown that when the spacing of "reference signatures" goes below roughly

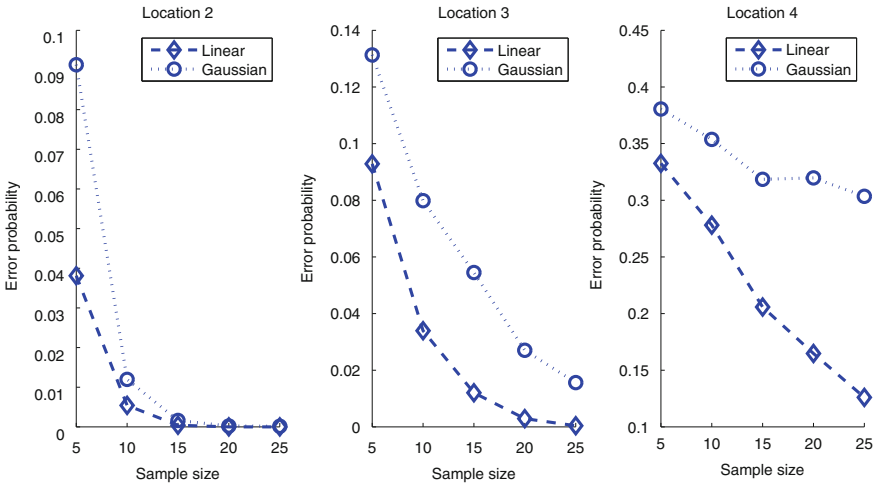


Fig. 8 Comparison of linear interpolation and the Gaussian model in terms of the error probability of GHT using the interpolated pdf

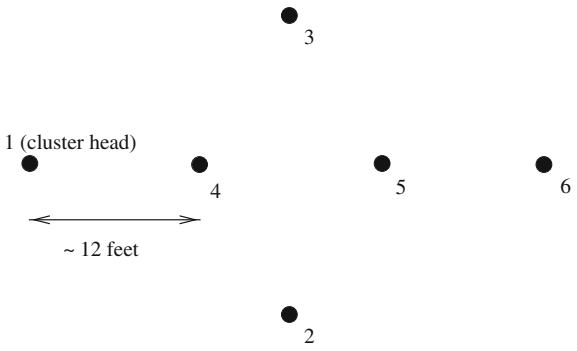


Fig. 9 Layout of the experiment. Node 1 is the clusterhead (listener)

10 m, the improvement in performance diminishes. The spacing of the "reference signatures" there is analogous to the distance between the two end-point locations in our pdf interpolation. The two end points in the working version (the "short") of our pdf interpolation happen to be a little more than 9 m (30 feet) apart. This result reinforces that of [10], as both indicate that taking empirical measurements at a spacial density of less than 9 or 10 m apart, or roughly 1 per 25 sq.m, carries diminishing benefit.

The interpolations above were done along a line. We also tested interpolations in a triangle, as shown in Fig. 9. The clusterhead (the listener) was placed at location 1 and the profile of location 5 was interpolated using those of location 2,3, and 6. Figure 10a shows the KL distance from the interpolated profile of location 5 to the empirical profile of itself and other locations. The figure shows that under linear

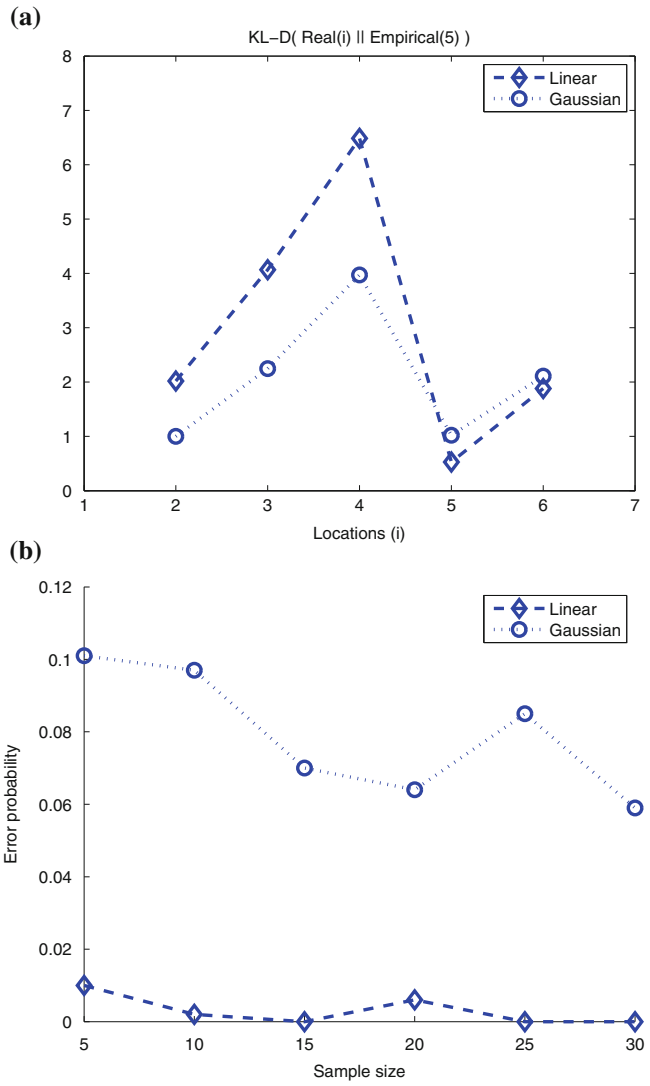


Fig. 10 **a** KL distances of the empirical pdfs to the interpolated pdf of location 5 (a clear minimum at 5 is desired) **b** Error rate of movement detection for location 5. Comparison of linear interpolation and the Gaussian model in terms of **a** the KL distance, **b** the error rate in GHT

interpolation, the distance to self is clearly lower than those to others—the plot has a clear minimum at location 5—while the distinction under Gaussian interpolation is less pronounced. Finally, Figure 10b shows the error probability of GHT using different profiles. Consistent with the KL distance, the linear interpolated profile out-performs the Gaussian one significantly.

5.3 Change of Signal Profiles Over Time

Consideration of short-term changes such as the effect of people walking around has already been built into our model. With that taken care of, we can consider the signal profiles to be relatively stable. In the long run, some re-profiling would be inevitable. Our experiences indicate that the profile experiences virtually no change for the time scale of one month, but re-profiling may be needed for a period beyond 4 months.

First, to assess the effect of an outdated profile on localization accuracy, we performed a three-part experiment that used a current profile, an one-month-old profile, and a four-month-old profile, respectively. When we used the one-month-old profile the probability of accurately reporting the correct landmark deteriorated by 2.27 % compared to the one obtained by using the current profile. On the other hand, when we used the four-month-old profile, the probability of accurately reporting the correct landmark deteriorated by 6.5 % compared to the one obtained by using the current profile. Real systems are typically overprovisioned so such a performance deterioration may be tolerable. Of course, the percentage drift in the corresponding probabilities of error was substantially higher but in any case we work with small probabilities of error. Furthermore, even when we make an error, the error is small because the mistaken location is a neighboring landmark in virtually all of the cases.

To gain more insight, we used again the KL distance to measure the drift of signal profiles. Signal profiles of four locations that resemble a neighborhood in our landmark graph were measured in 2010, first in July, then in the beginning of November (i.e., after 4 months), and then at the end of November. Table 2 shows that the KL distances of signal profiles between pairs of locations are on the scale of 3–8. On the other hand, Table 1 shows that, the signal profiles changed by a KL distance of around 0.5 from the beginning to the end of November, which is negligible compared with the pairwise distances. However, the maximum drift in KL distances from July to November is around 3, which is large enough (compared to the pairwise distances) to affect the localization results.

Table 1 KL distance of signal profile drifting

	KL-D Early Nov vs. Late Nov	KL-D July vs. Early Nov
Location 1	0.54	0.57
Location 2	0.68	2.15
Location 3	0.58	1.12
Location 4	0.40	3.19

Table 2 KL distance between locations

KL-Distance	Location 1	Location 2	Location 3	Location 4
Location 1	–	4.42	6.44	6.49
Location 2	3.29	–	5.17	6.96
Location 3	6.86	5.48	–	3.41
Location 4	8.01	7.87	3.45	–

6 Conclusion

This chapter reviews a set of carefully designed rules and algorithms that underpins a successful two-tier wireless RSSI-based positioning system. Both theoretical and experimental justifications are provided. In addition to working out the details of the design, the experimental validation of the formal pdf interpolation and that of the single-clusterhead-based movement detection are valuable pieces of information to practitioners.

Our approach to the localization problem is formal. The coverage area is modeled as a landmark graph, whose nodes and edges are locations of interest. The signal profiles between locations are processed using a rigorous pdf interpolation technique. The decision rules are GLRT and GHT, which provide error probability estimates. And these error probability estimates are used in a MILP formulation to come up with optimal system deployment.

Furthermore, these formal approaches are validated by experiments. We showed that the overall accuracy of our system compared favorably against other state-of-the-art methods using the same low-cost hardware (although we recognize that the conditions of experimentation are not strictly comparable). The idea of combining localization and movement detection to improve system efficiency is also shown to be practical, i.e., an efficient deployment solution that satisfies the needs of both localization and movement detection does exist. In addition to validating the entire system, we also zoomed in to the particularly interesting pdf interpolation technique. In these more focused experiments, we not only showed that this theoretically appealing interpolation technique actually works, but also showed that the reason why we expected it to work is also valid, i.e., the interpolated pdf carries more information (as measured by the KL distance) for distinguishing between locations.

It is worth noting that the use of advanced decision theory does not make our approach more difficult to use; quite the contrary. We do record full pdf information and use formal optimization to decide clusterhead placement, which makes the initialization of the system a little more complex than other methods. However, after implementing these algorithms, deploying the system is very easy exactly because the approach is "formal" at every step. There is little need for trial-and-error type of adjustments. In short, it is our hope to contribute to the rational decision making process in constructing a localization system.

References

1. R.M. Estanjini, Y. Lin, K. Li, D. Guo, I.C. Paschalidis, Optimizing warehouse forklift dispatching using a sensor network and stochastic learning. IEEE Trans. Industr. Inf. **7**(3), 476–486 (2011)
2. J. Caffery, G. Stuber, Subscriber location in CDMA cellular networks. IEEE Trans. Veh. Technol. **47**(2), 406–416 (1998)
3. A. Weiss, On the accuracy of a cellular location system based on RSS measurements. IEEE Trans. Veh. Technol. **52**(6), 1508–1518 (2003)

4. R. Want, A. Hopper, V. Falcao, J. Gibbons, The active badge location system. ACM T. Inform. Syst. **10**, 91–102 (1992)
5. N.B. Priyantha, A. Chakraborty, H. Balakrishnan, The cricket location-support system, in *Mobile Computing and Networking*, 2000, pp. 32–43. "http://citeseer.nj.nec.com/priyantha00cricket.html"
6. S. Tarzia, P. Dinda, R. Dick, G. Memik, Indoor localization without infrastructure using the acoustic background spectrum, in *Proceedings of the 9th international Conference on Mobile Systems, Applications, and Services (MobiSys)*, 2011, pp. 155–168
7. I. Guvenc, C. C. Chong, F. Watanabe, NLOS identification and mitigation for UWB localization systems, in *Wireless Communications and Networking Conference (WCNC 2007)*, March 2007, pp. 1571–1576
8. N. Patwari, S. Kasera, Robust location distinction using temporal link signatures, in *Proceedings of the 13th Annual ACM International Conference on Mobile Computing and Networking (MobiCom '07)*, 2007, pp. 111–122
9. P. Bahl, V. Padmanabhan, RADAR: An in-building RF-based user location and tracking system, in *Proceedings of the IEEE INFOCOM Conference Tel-Aviv*, Israel, March, 2000
10. K. Lorincz, M. Welsh, Motetrack: A robust, decentralized approach to RF-based location tracking, in *Springer Personal and Ubiquitous Computing, Special Issue on Location and Context-Awareness*, 2006, pp. 1617–4909
11. K. Kaemarungsi, P. Krishnamurthy, Modeling of indoor positioning systems based on location fingerprinting, in *Proceedings of the IEEE INFOCOM Conference*, 2004
12. J. Hightower, R. Want, G. Borriello, *SpotON: An indoor 3d location sensing technology based on RF signal strength*, University of Washington, Department of Computer Science and Engineering, Seattle, WA, UW CSE 00–02-02, February 2000
13. P. Castro, P. Chiu, T. Kremenek, R. Muntz, A probabilistic location service for wireless network environments, in *Proceedings of Ubicomp*, Atlanta, GA: ACM, September 2001
14. J. Krumm, E. Horvitz, Locadio: Inferring motion and location from Wi-Fi signal strengths", in *First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (Mobiquitous 2004)*, 2004, pp. 4–13
15. K. Chintalapudi, A. Padmanabha Iyer, V. Padmanabhan, Indoor localization without the pain, in *Proceedings of the Sixteenth Annual International Conference on Mobile Computing and Networking.* ACM, 2010, pp. 173–184
16. N. Patwari, A.O. Hero, M. Perkins, N.S. Correal, R.J. O'Dea, Relative location estimation in wireless sensor networks. IEEE Trans. Signal Proc. **51**(8), 2137–2148 (2003)
17. K. Yedavalli, B. Krishnamachari, S. Ravula, B. Srinivasan, Ecolocation: A sequence based technique for RF-only localization in wireless sensor networks, in *The Fourth International Conference on Information Processing in Sensor Networks* (Los Angeles, CA, April, 2005)
18. I.C. Paschalidis, D. Guo, Robust and distributed localization in sensor networks, in *Proceedings of the 46th IEEE Conference on Decision and Control* (New Orleans, Louisiana, December, 2007), pp. 933–938
19. I.C. Paschalidis, D. Guo, Robust and distributed stochastic localization in sensor networks: Theory and experimental results. ACM Trans. Sensor Networks, **5**(4), 34:1–34:22, 2009
20. M.A. Youssef, Collection about location determination papers available online, 2008, http://www.cs.umd.edu/moustafa/location_papers.htm
21. M. Kjærgaard, A taxonomy for radio location fingerprinting. Lect. Notes Comput. Sci.: Location-and Context-Awareness **4718**, 139–156 (2007)
22. C. Chang, A. Sahai, Cramer-Rao-type bounds for localization. J. Appl. Signal Process. **2006**, 113 (2006)
23. S. Gezici, A survey on wireless position estimation. J. Wireless Pers. Commun. **44**(3), 263–282 (2008)
24. F.H. Bursal, On interpolating between probability distributions. Appl. Math. Comput. **77**, 213–244 (1996)
25. S. Ray, W. Lai, I.C. Paschalidis, Statistical location detection with sensor networks. Joint special issue IEEE/ACM Trans. Networking IEEE Trans. Inf. Theory **52**(6), 2670–2683 (2006)

26. T. Cover, J. Thomas, *Elements of Information Theory* (Wiley, New York, 1991)
27. T. Cover, J.A. Thomas, *Elements of Information Theory* (Wiley, New York, 1991)
28. W. Hoeffding, Asymptotically optimal tests for multinomial distributions. Ann. Math. Statist. **36**, 369–401 (1965)
29. A. Dembo, O. Zeitouni, *Large Deviations Techniques and Applications*, 2nd edn. (Springer-Verlag, New York, 1998)
30. M. Daskin, *Network and Discrete Location* (Wiley, New York, 1995)
31. F. Özsoy, M. Pınar, An exact algorithm for the capacitated vertex p-center problem. Comput. Oper. Res. **33**(5), 1420–1436 (2006)

Chapter 6
Protocol Design for Real-Time Estimation Using Wireless Sensors

Yaser P. Fallah and Raja Sengupta

Abstract This chapter discusses how a wireless sensor network can be built for real-time estimation purposes. The finite capacity of a wireless network in delivering information means that a real-time estimation process has finite accuracy too. Improving accuracy requires faster sampling and more communication; however, faster communication is not always a possibility due to scalability issues and the limited capacity of a network. In fact, in networks where the wireless medium is shared amongst many nodes, the increase in the amount of communication may even have a negative impact on the capacity of wireless medium, and in turn on the quality of real-time estimation (e.g., loss of network capacity as seen in CSMA/CA—Carrier Sense Multiple Access/Collision Avoidance- networks). In this chapter we describe methods and protocols that can be employed to control the behavior of nodes to allow maximal use of the shared medium for the purpose of real-time estimation. We describe how transmission control protocols (adapting rate and range of communication) should be applied in a wireless network of mobile sensors, such as vehicles, to allow the highest possible accuracy given the limitations of the medium. Such protocols have been evaluated in the form of transmission rate and range control schemes in wireless vehicular networks, with the purpose of real-time vehicle position tracking.

Y. P. Fallah (✉)
West Virginia University, Morgantown, USA
e-mail: Yaser.Fallah@mail.wvu.edu

R. Sengupta
University of California Berkeley, Berkeley, USA
e-mail: sengupta@ce.berkeley.edu

H. M. Ammari (ed.), *The Art of Wireless Sensor Networks*,
Signals and Communication Technology, DOI: 10.1007/978-3-642-40066-7_6,

1 Introduction

Wireless Sensor Networks (WSN) may be used for sensing and observing a physical phenomenon in real-time. Such an application may be referred to as real-time tracking or estimation over WSNs. Tracking latency and resolution will be directly dependent on the dynamics of the sensed phenomenon and the capacity of the WSN. The requirements of timeliness and accuracy are usually high when real-time estimation is considered. For example, in a mobile sensor network formed by vehicles, where each node is expected to estimate the states of its neighboring vehicles, the requirements become very stringent [1, 5]. This is in particular true when applications like collision warning are considered. Given vehicle dynamics and movements, state of a vehicle (e.g., its speed, position, heading, etc.) needs to be tracked with a latency of less than a few hundred milliseconds. Achieving this level of real-time tracking accuracy puts a huge burden on the wireless network that is the communication backbone of the vehicular mobile sensor network. The main issue is the limited capacity of the network, and the effect of the sensor node loads on this capacity limit.

The issue of limited capacity has always been one of the main constraints of networks in general, and wireless networks in particular. However, in wireless networks where coordinated channel access is impossible or difficult, the issue becomes compounded with collisions and wasted bandwidth due to uncoordinated access. This issue was in particular observed for vehicular mobile sensor networks [6, 15, 31, 33].

Dealing with limited channel capacity has been one of the main topics of research from the early days of the conception of telecommunications. Traditionally, two approaches have been considered for addressing this issue. One approach is to try and compress the information in the application layer in order to fit more information into a limited size communication pipe. The other approach is to fit as many bits as possible in a given bandwidth, which can be seen as increasing the available capacity (size of the communication pipe). While these approaches have their own merits and are valid efforts, their usefulness and validity is limited to cases where application and channel (communication network) behavior are independent and can be studied separately. This is usually true when a single communication channel between two nodes is considered; however, in networks where the communication medium and intermediate devices are shared amongst many nodes, the interdependence of the application and network behavior becomes important. An early example of this issue was the problem with Internet congestion which would render the entire network useless and drive the capacity to zero [43]. Mitigating such issues resulted in coarse congestion management schemes to be adopted in transmission control protocols like TCP. However, while TCP is designed to be agnostic to the application data types, knowledge of the application will yield considerable benefits when the interdependence of application and network behaviors are concerned [12].

Wireless sensor networks are indeed meant for sensing and tracking purposes. It is, therefore, reasonable to look for methods of employing the knowledge of the interaction between sensing and tracking applications and the network in order to achieve the best possible results. Such considerations are in fact the basis of some of the most

recent developments in devising methods for real-time estimation of vehicle states over mobile sensor networks [11, 14]. This chapter will use the vehicular application as a case for studying how communication control protocols can be designed with the purpose of enabling real-time estimation applications. The process of designing such a protocol is described in the next sections, following an introduction to the tracking application that relies on them.

2 Real-Time Estimation and Tracking Over a Multi-Access Channel

An example of a mobile sensor network that is used for tracking real-time processes is that of a vehicular network for cooperative safety [1, 5]. Such a network is comprised of mobile nodes (vehicles) that sense their state (position, speed, etc.) using GPS and onboard navigation sensors (e.g., gyro's and accelerometers) and broadcast the information over an 802.11p network [2–4]. Vehicles receiving the broadcast messages will try to create a real-time map of vehicles in their neighborhood, tracking each vehicle in their vicinity (Fig. 1). A hazard detection algorithm continuously analyzes the map to detect dangerous situations, and warn the driver [1] or take an evasive action. Given the criticality of the application, it is very important that tracking of other vehicles is performed with adequate accuracy (for example less than 1 m error with confidence of 95 %). This application is called Cooperative Vehicle Safety (CVS) and is being planned for large scale deployments in the coming years (under USDOT Connected Vehicle program).

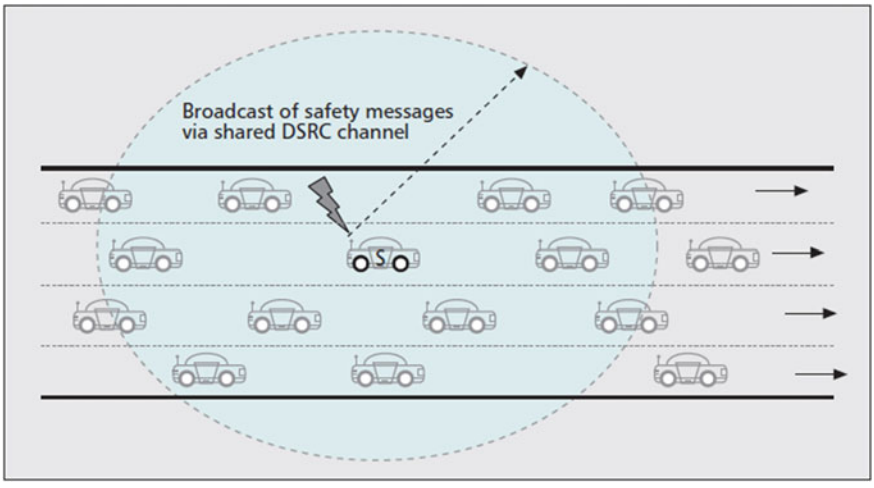


Fig. 1 Cooperative vehicle safety is enabled by a mobile sensor network consisting of vehicles sensing and broadcasting their state information over an 802.11p wireless network

The CVS network can be described in a simple problem where multiple Linear Time Invariant (LTI) dynamical systems track each other over a shared channel. Each LTI dynamical system represents a single vehicle; each vehicle tries to track all its neighboring vehicles up to a certain distance. While we describe the LTI example of a vehicle movement, other dynamical systems may also be used in the same problem setting and formulation with straightforward enhancements. In this section, we describe a mathematical framework for the tracking problem and compare the performance of different communication policies in ideal conditions. In subsequent sections we consider more complex and real communication media and re-evaluate the tracking performance for different algorithms. For a decentralized communication policy, it is shown that the effect of tracking application and underlying network on each other has to be considered to achieve robust tracking performance.

2.1 Tracking Problem Formulation

Estimation and tracking over a communication channel has been classically formulated as a sender–receiver pair with one-to-one channel setting (e.g., in [8]). Simple channel loss models are usually considered in such formulations, assuming that channel loss follows a stationary distribution and is independent of the transmission behavior at the sender. Such assumptions make the problem analysis tractable, but discount the very important effect of sender behavior on wireless network performance, in particular when multiple nodes share the channel (i.e., multi-access channels, which are commonplace in WSNs). As a result, the classical setting is not suitable for multi-access channels because the performance of such channels heavily depends on the transmission behavior of all nodes. For example, an increase in transmission attempts does not always translate to an increase in the amount of information successfully delivered. In certain cases, like in simple Aloha networks, the capacity may actually decrease. Knowing this fact, we formulate the problem as follows:

Consider a real-time tracking problem with finite n dynamical systems $n = 2,3, \ldots$, where their dynamics are assumed to be decoupled. In analogy to the CVS problem setting, these nodes represent all vehicles in proximity of a given vehicle. To ease our discussion, for each node index $j \in \{1, 2, ..., n\}$, let the state transition be represented by a simple LTI model. We have

$$x_j(t) = a_j \times x_j(t-1) + b_j \times u_j(t-1) + \varepsilon_j(t-1) \tag{1}$$

where $x_j(t)$ is the state of node j, $u_j(t)$ and $\varepsilon_j(t)$ are the stationary zero-mean input and noise processes with bounded variance, and t is the time index $t \in$ N. In case of a vehicle, a_j and b_j represent the mechanical characteristics and physical laws that govern vehicle j, respectively. The amplification factor a_j decides how fast the state x_j evolves. In analogy to our CVS problem setting, by extending the dimensions of parameters in the LTI model (1), a_j can be used to model physical laws that

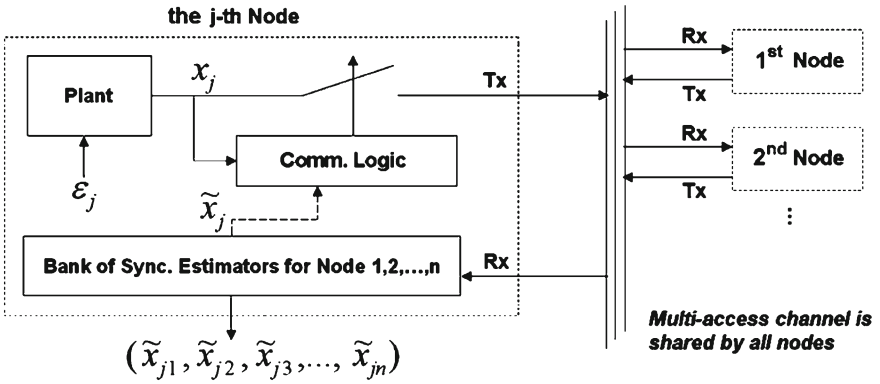


Fig. 2 Node internal structure in the analyzed real-time tracking problem

govern that vehicle's movement, and $\varepsilon_j(t)$ can be used to model mechanical and driver related disturbance [17]. Such an LTI approximation has been used in control literature for example in [7] and [37].

The internal communication and estimation processes of a vehicle can be represented as in Fig. 2 ([17]). The n nodes depicted in the figure share the same wireless channel, for example the 802.11p or DSRC—Dedicated Short Range Communications-channel [2–4]. Each node contains a discrete-time LTI scalar process as in (1), a communication logic, and a bank of synchronized model-based estimators that are used to track other vehicles in the neighborhood.

In this setting, the true state (a real number with an acceptable distortion) of sender j will be *broadcast* to the shared channel by its communication logic. The communication logic has to make two main decisions regarding broadcast of the state information. One is to decide when to transmit a message, the other is to decide to what distance transmit it. The former corresponds to determining the policy of sampling and communication of samples to individual receivers and is sometimes called "rate control"; the latter concerns with the network size and range of receivers of the state information, constituting the "range or power control" scheme. The rate and range control mechanisms can be together seen as the transmission control protocol for multi-access channels, just as TCP is the transmission control protocol in IP-based point to point connections.

At the receiver side of the messages, an estimation process is run to recover the state of the sending vehicle. The model-based estimator at receiver i operates on a discrete clock and performs an estimation action at each estimation time, based on whether a new message is received or not. Therefore the estimator switches between the following two modes:

- If no information regarding node j is received at $t-1$, use previous estimate at $t-1(X_{ij}(t-1))$ and the known model (1) to carry on and derive the new estimate: $X_{ij}(t) = \alpha j \; . \; X_{ij}\;(t-1)$

- If state information of j is received at $t-1$, $x_j(t-1)$, use it to reset estimation error at $t-1$, then use the model to estimate the state at time t: $X_{ij}(t) = a_j.x_j(t-1)$

Note that the state information received at time t over the wireless network is always representing the sender's state at some t' time where $t' < t$ (to account for communication delay). For simplicity assume $t' = t - 1$. Therefore, the actual state at any given time t has to be recovered using the estimation process.

In the tracking problem described above, the arrival of a new measurement from sender j to receiver i resets the tracking process, by setting the estimated state to that received from the sender. Between message arrivals, the real-time tracking error could grow due to noise and disturbance. Note that the above calculation of $X_{ij}(t)$ is a minimum mean squared error (MMSE) optimal estimate if we assume $u_j(t)$ and $\varepsilon_j(t)$ have zero mean. The tracking error $e_{ij}(t)$, the ith vehicle's estimation error toward vehicle j, is defined as $e_{ij}(t) - X_{ij}(t)$. Due to the hidden node problem or channel fading, this tracking error might vary for different nodes.

2.2 Communication Policies: Uncontrolled versus Controlled

Communication policies that perform the tracking task without dynamically adjusting the process to network or tracking performance are referred to as "uncontrolled" policies [32]. In contrast, communication methods that control the behavior of the communication logic based on the performance of state estimation process or the network are called controlled policies [32]. A number of controlled and uncontrolled policies for simplified networks such as slotted ALOHA have been described in [17]. An interested reader can find the details of these methods in [17]; however, for the sake of clarity, sample results comparing the performance of three different representative classes of methods over slotted Aloha network is shown in Fig. 3.

In Fig. 3, the estimation error is found versus |a|, a parameter determining the speed of change for the tracked process. It is observed that probabilistic method, which is an uncontrolled method based on random transmission of messages at discrete time instances, is unable to produce stable results for |a| > 1; while a controlled method

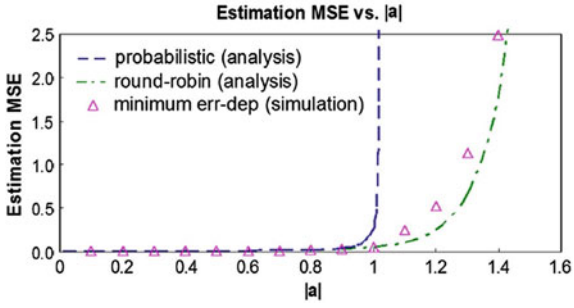


Fig. 3 Estimation MSE for controlled policy (minumum err-dep), uncontrolled policy (probabilistic), and ideal scheduled policy (round robin)

(minimum err-dep: controlling rate of transmission based on detected collisions) comes close to reaching the capacity of the channel for supporting the tracking application, which is seen for the "round robin" method. The round robin method is based on ideal, centralized deterministic, and scheduled access to the channel and periodic transmission of messages (which is not practical). While the difference of controlled and uncontrolled methods are analytically examined and demonstrated for simplified and idealized cases in Fig. 3 (from [17]), the complexities of multi-access schemes such as CSMA/CA used in 802.11p or other wireless networks, require a careful analysis before a proper design is possible. The next sections describe a methodical way of analyzing the different components of the system, i.e., real-time tracking process and the network, and provide a systematic approach to designing the communication protocols for the purpose of achieving robust real-time tracking over WSNs.

3 Transmission Control Protocol to Support Real-Time Tracking

It was shown in the previous section that controlled communication policies can yield great benefits in real-time tracking over simple multi-access networks. It has been shown in several recent research works by authors [14, 16] that such benefits are even greater when more complex wireless networks such as 802.11p (DSRC) are considered. The benefits come from the fact that the tracking process is affected by, and does affect, the network performance. In this section, the designs of transmission control policies that support real-time tracking as their main application are considered. The design is specifically studied from the perspective of controlling the main two decision parameters, range (d) and rate (time and frequency r) of state broadcast. Controlling the rate and range of transmission is primarily done to support the real-time tracking process, which is the main process of applications like CVS. In an analogy to Internet Protocols, the combined control schemes for rate and range is in fact a "transmission control protocol," acting much like TCP that controls the transmission of messages over Internet Protocol (IP). In the following subsections we first examine the general topic of transmission control and also survey some existing mechanisms for transmission control in wireless sensor networks. The discussion is then directed towards an analysis of the multi-access networks, on which systems like CVS operate, in order to help readers understand the design of transmission control protocols for tracking over multi access channels.

3.1 Transmission Control Protocols

Transmission control is mainly performed to improve the performance of the "application" that uses the underlying communication and networking services. Such improvements may be achieved indirectly through merely improving reliability and

capacity of the underlying network, or by controlling the communication in a way that directly impacts the application. For example, a congestion control scheme may be used to improve the capacity and generally improve application performance; whereas communication timing may be controlled to directly impact how a tracking application works (as we will see later in this chapter). Therefore, transmission control protocols can be viewed as application specific or generic algorithms. Protocols such as TCP fall under the generic category.

With the introduction of wireless data communication, TCP was observed to be lacking the features needed for communication over wireless networks. The main issue was that the congestion control mechanism of TCP would mistake wireless channel loss for congestion and reduce the transmission rate, further harming the application performance [18]. As a result, several schemes were suggested to resolve the issue (a comparison of these schemes is found in [19]). Even though many of these schemes do enhance the performance, no single scheme has found widespread use in all wireless sensor network applications due to the fact that underlying wireless networks are very diverse and the main TCP objectives (higher throughput and reliability) are not directly aligned with many specialized application requirements in WSNs.

A good survey of different transmission control (transport) protocols for wireless sensor networks is found in [20]. The main functionality provided by the various transport protocols for WSNs are end-to-end reliability and congestion control. In the application considered in this chapter, real-time estimation over broadcast multi-access channels, the issue of end-to-end reliability is replaced by "broadcast reliability". Congestion control, however, is directly needed as it also impacts broadcast reliability through improving network performance.

Congestion control in any type of network, including WSNs, requires some method of detecting congestion. For example, algorithms like Sensor Transmission Control Protocol (STCP [21]), Fusion [22], and Congestion Detection and Avoidance (CODA [23]) rely on queue length measurement in intermediate nodes on a multi-hop path of the data. Algorithms such as Congestion Control and Fairness (CCF [24]) and Priority-based Congestion Control Protocol (PCCP [25]) infer congestion at the end nodes. CCF infers congestion based on packet service time, whereas PCCP calculates congestion degree as the ratio of packet inter-arrival time and packet service time. The response to congestion varies for different algorithms. Fusion controls congestion in a start–stop flow control fashion by making neighboring nodes stop forwarding packets to the congested node upon congestion detected and notification. CODA and STCP adjust sending rate in an Additive Increase Multiplicative Decrease (AIMD) way; AIMD is also the way TCP manages sending rate of data at the source. PCCP and CCF work based on exact adjustment of the Hop-by-Hop rate (compared to relative methods like AIMD).

While these transport protocols target reliable delivery and congestion control as the main objectives, their use with the real-time tracking application will not be directly possible due to several characteristics of the tracking application. First characteristic is the broadcast nature of the underlying communication topology (no multi-hop); second is the fact that real-time tracking in systems like CVS

requires both accuracy of tracking for individual nodes and an acceptable distance up to which tracking should be done accurately. The second feature brings in the dimension of space or range of transmission, in addition to the rate or timing of message transmission. Nevertheless, the existing rate adjustment concepts can be useful in the design of transmission control protocols in our case. The main difference remains the additional dimension of range in this case. With this understanding we next review how transmission control for our application of interest (tracking over multi-access channels of mobile sensor networks) can be done.

3.2 Transmission Control to Support Real-Time Tracking in Broadcast Mobile Sensor Networks

To understand what may be required for adapting rate and range of node (vehicle) state broadcast, the CVS network performance should be characterized in terms of different choices of rate and range. This will also allow for finding methods of congestion detection. Deriving a performance model for the CVS network is in fact a complicated task, primarily due to the fact that the network is not a traditional one. On one hand, it is based on a MAC layer that follows the CSMA/CA protocol. On the other hand, the hidden node phenomenon causes many nodes that are not in range of each other to behave almost in an ALOHA fashion. Several other phenomena such as silencing effect and prolonged collision ensue from such combination of CSMA/CA and ALOHA-like networks [11]. In [11], these issues are examined in depth and great detail. In this section a brief overview of the model is presented, with the purpose of deriving the network performance characteristics, as a function of the rate and range of transmission.

3.2.1 Network Analysis

To quantify the effect of controllable parameters (rate r and range d) on the overall performance of the real-time tracking process in a CVS network, we need to relate the performance of the broadcast network to the performance of the tracking application. For this purpose, a model describing the accuracy of the tracking process at different neighboring nodes as a function of system parameters is needed. Designing such a model requires modeling the interaction of the estimation processes in the application layer with the multiple access scheme. Such an analysis is possible when very simple networks (like ALOHA) and simple communication/tracking policies (like random message generation) are considered [17]; however, the problem becomes much more complex when the CSMA/CA network and communication policies of CVS are analyzed. Considering the significant effect of hidden nodes and complexities it creates for network modeling, such an integrated model has proven to be intractable so

far. While efforts exist that try to empirically model such systems [12], a mathematical framework has not been presented yet. As a result, we have to use an intermediate performance measure from the CVS network that corresponds to CVS performance. We emphasize that such measures are used only to gain insight into the design of control schemes, and not to, for example, formulate optimization schemes that focuses the solution on better network performance in place of system performance. Given the well-known fact that a higher rate of message arrival at an estimation process will result in more accurate predictions [7, 32], it is plausible to rely on a corresponding network performance measure as a substitute of the more complex overall performance measure in analyzing the system (rather than in formulating final solutions).

The network performance measure should consider the fact that in a CVS broadcast network both rate of packet reception at receiver vehicles and number of receivers are important; that is, it serves the objective of CVS to include more neighbors in the transmission range, up to some maximum range (e.g., 300 m).

A straightforward representation of this performance measure is a form of the broadcast packet delivery rate, defined here as Information Dissemination Rate (IDR). IDR represents the number of copies of a packet delivered per unit time from a single vehicle to its neighbors up to a given distance d_{max}:

$$IDR = \sum_{i \in Set\ of\ Neighbors\ upto\ d_{\max}} r.P_{suc}(i) \tag{2}$$

where $P_{suc}(i)$ is the probability or ratio of successful message delivery to each node i, and r is the rate of message transmission by each node. This definition can be modified by including a weighting function that gives a higher weight to closer vehicles (if CVS design assumes nearer nodes are more important and require higher accuracy, thus higher reception rate). Here we focus on normal IDR for clarity of discussion. Weighted IDR is similarly treated.

IDR is expected to increase with the increase in transmission rate or range, if such increases do not lead to congestion and increase in packet drop. However, the increase in rate or range significantly and adversely affects the performance of the network MAC layer, which follows the CSMA/CA protocol. In addition to normal packet losses due to CSMA/CA collisions, the effect of hidden node interference is also a significant factor in a CVS. In particular, when the near 1-D topology of a highway is considered, hidden node collision becomes the dominant source of packet loss [14, 29].

To quantify how parameters r and d affect IDR, an extensive set of simulation experiments and detailed mathematical analysis were performed [11, 12]. The simulation and mathematical analysis results have been compared and verified in [11]; due to space limitation the mathematical model could not be included here. However, we report the result from [11, 12] to the level needed for the design of the transmission control protocol of this chapter.

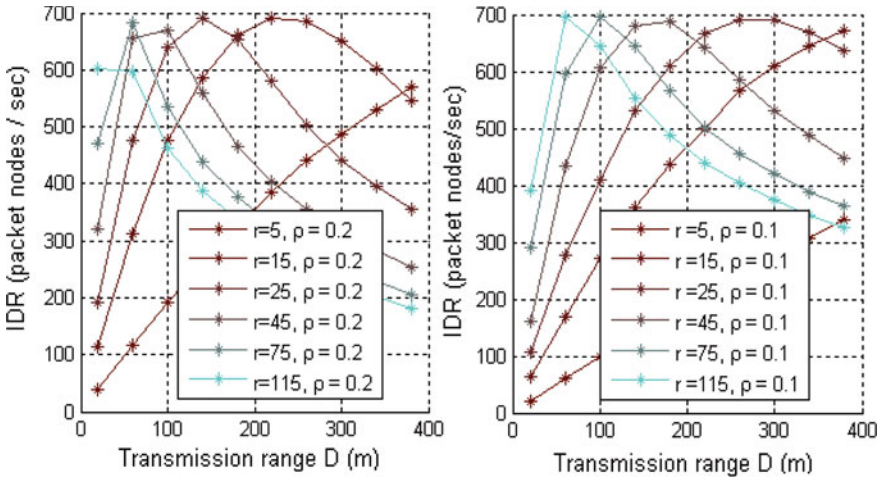


Fig. 4 IDR versus range of transmission d for different rate r and density ρ

The simulations reported here have been performed using an event driven 802.11 MAC simulator developed at UCB (for static topology and initial study) and using OPNET (for mobile nodes and final evaluation) considering the 802.11p default parameters [3], with the bitrate of 3 Mbps (similar to 6 Mbps with 50 % duty cycle of 1609). We have considered a 3 Km long stretch of a highway (results collected from the middle 1 Km of the topology to avoid edge effects). Nodes were placed randomly at intervals uniformly distributed in a $(0, 2\rho)$ range, where ρ is the traffic density in vehicle/meter. While in the study presented in this subsection node locations were fixed (to more precisely observe the MAC performance in relation to r and d parameters), in later evaluation experiments we used mobile nodes based on trajectories generated by SHIFT traffic simulator [41].

In the first set of experiments we observe IDR, and plot it versus the choice of transmission range d for different values of r. Simulation results for two typical values for road density ρ are shown in Fig. 4. To generalize Fig. 4 and better observe the changes in IDR for different choices of rate and range, the model in [11] is used to plot a 3-D graph of IDR versus r and d at high resolutions, for a given value of density ρ. The result is shown in Fig. 5. Replacing range d with ρ in this figure results in a similar plot, since for a highway scenario with 1-D distribution of vehicles the two parameters are interchangeable through a linear transformation.

As it is observed in these figures, the maximum level of IDR is reached at a different value of range d, for any given choice of r and ρ. This fact further emphasizes the need for adaptation of rate or range of transmission. Another interesting observation from these figures is that the maximum value of IDR is the same for all curves in Fig. 4. Similarly, in Fig. 5, the maximum value of IDR is seen to be happening not at a single point, but on a curve. Many choices exist for *(r, d)* pairs that generate maximum IDR. These pairs fall on a curve that forms the elongated peak of the

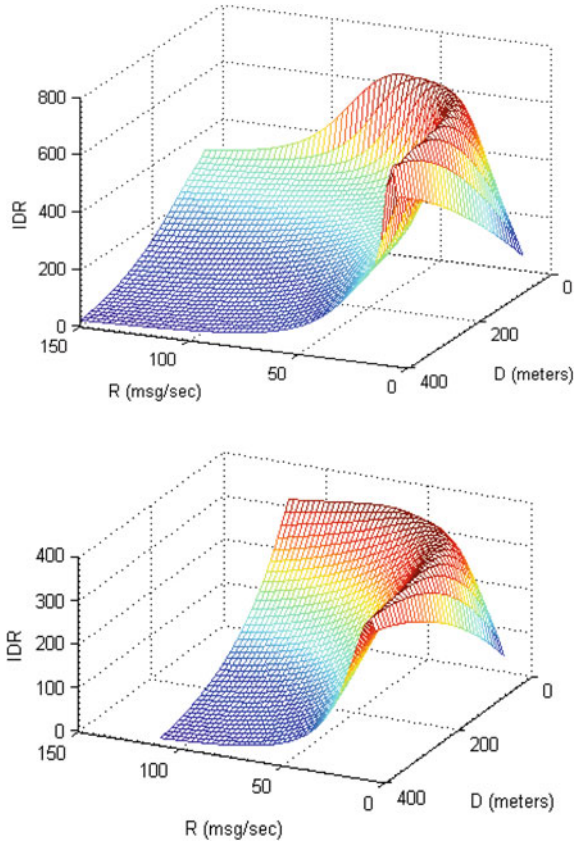


Fig. 5 *IDR* versus *R* and *D*, *Top* seeing only the MAC performance, *Bottom* considering both MAC and PHY, [11]

plot in Fig. 5. This observation leads us to believe that maximization of IDR can be achieved by adjusting one of the parameters of r or d, if other constraints allow. Such an observation relaxes the 2-D problem of controlling r and d to a one-dimensional optimization problem if one parameter is in an acceptable range.

3.2.2 Congestion Control to Improve Broadcast Performance

The observation from Fig. 5 leads us to the conclusion that congestion mitigation for the purpose of improving broadcast throughput (i.e., IDR) can happen by changing either rate or range of transmission. However, not all pairs of rate and range that maximize IDR are acceptable; for example, a very small value of range and a large value of rate is not useful since it does not allow coverage in a desired neighborhood which may require a larger range. Similarly, a large range and small rate will lead to inaccurate tracking in a very large neighborhood, which will still be unacceptable. Therefore, a congestion control design that relies on network performance (IDR)

maximization, should also consider other restrictions imposed by the application. Such restrictions in the case of CVS are neighborhood range and tracking accuracy. Considering this fact, the design of a transmission control protocol can be broken into two separate parts, focusing on maintaining one parameter in an acceptable (or optimal from CVS perspective) range, while using the other one to maximize IDR. The goal of the protocol will be to achieve desirable system performance, which should be seen from the standpoint of "real-time tracking or estimation." We approach this problem by designing a rate control method that aims for achieving the required tracking accuracy, and then using the range parameter in conjunction with the result from rate control to increase the reach of the messages and improve the number of tracked nodes. The next two sections describe how such a design is accomplished. The overall design of the transmission control protocol is described following the discussion on how individual components (rate and range adaptation) are designed.

4 Controlled Policy for Transmission Rate Adaptation

In this section we consider the problem of controlling communication time and frequency with the purpose of maximizing the efficiency of the estimation process over a network. Here, efficiency can be defined as achieving higher accuracy at lower communication cost. There are two main categories of rate control for tracking purpose. One is to rely on periodic transmission of messages at a given rate [5], and then adapting the rate to network conditions [38]. The other method is to use an error-dependent policy where transmission of messages is directly controlled by an estimate of the performance of the tracking process and potentially the network condition [17]. The latter produces a flow that is inherently variable rate, but its overall average output rate can be adjusted by controlling how sensitive the algorithm is to tracking error.

While both methods can be tuned to generate the same average rate of message generation, and potentially achieve the same effect on the network, their real-time tracking performance will be greatly different. The periodic transmission is in a way an uncontrolled method when we consider the tracking process as the controllable entity. The error-dependent policy on the other hand is a controlled method. To illustrate the difference between the two methods, consider the simple trajectory of Fig. 6.

Here, periodic transmission of vehicle state information corresponds to periodically sampling and sending the samples. The figure shows five equally spaced samples. At the receiver, a model-based estimator (e.g., following a first-order Kinematic model) is used to recover the trajectory of the sender vehicle for the time between samples. It is seen that the estimation error grows if vehicle movement model changes from the sampled model. For example, if the sampled information includes position and speed vectors, and a constant speed model is used to estimate the position in between sampling times, any speed change (acceleration) or direction change during

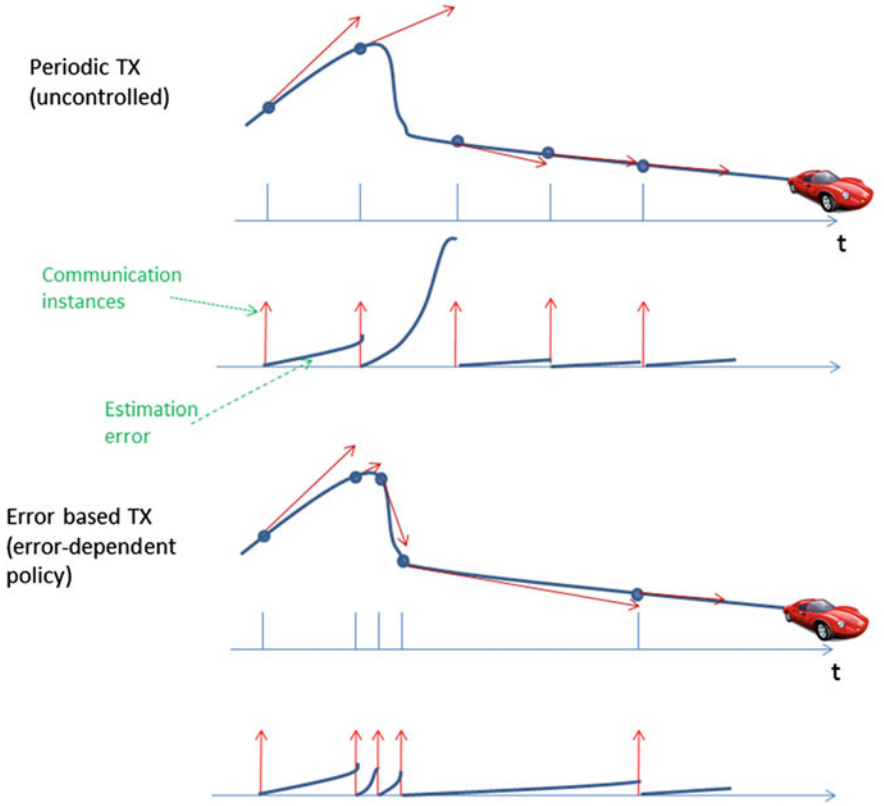


Fig. 6 Error-dependent policy (*bottom*) versus periodic or uncontrolled policy (*top*). Both methods are using the same overall message rate (five messages in the time window shown)

the time between samples causes the model to become wrong (since speed would be different). A wrong model then creates an error that grows with time. The error is particularly high when change in the model is significant (e.g., seen in the form of direction change in Fig. 6).

To counter this issue, the five samples could be redistributed so that they concentrate more on the area of the trajectory where there are sudden changes. The short term rate of messages will vary, producing higher rate where there is concentration of messages and lower rate where there is no change in trajectory. However, the overall rate over the shown time window remains the same for both algorithms. By concentrating communication samples around times when there are large changes in vehicle state, we are able to considerably reduce the tracking error [6, 31]. The reason is that the estimation process which works based on a constant speed model, will receive more updates when its error may become high due to change in the model parameters (speed, heading). At times when the trajectory deviates very little from

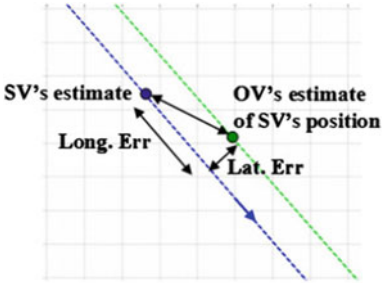


Fig. 7 Error must be calculated for longitudinal and lateral components [6]

the path estimated by the model-based estimator, there will not be need for new messages, since they do not considerably improve the estimation accuracy.

Realizing the error dependent policy requires that the sender have some indication of the estimation error (of the sender's state) at the receiver. This is achieved by including a replica of the receiver node's estimator of the state of the sender in the sender itself. This replica is called "remote estimator" in Fig. 8, and replicates the "neighbor estimator" in the receiver node.

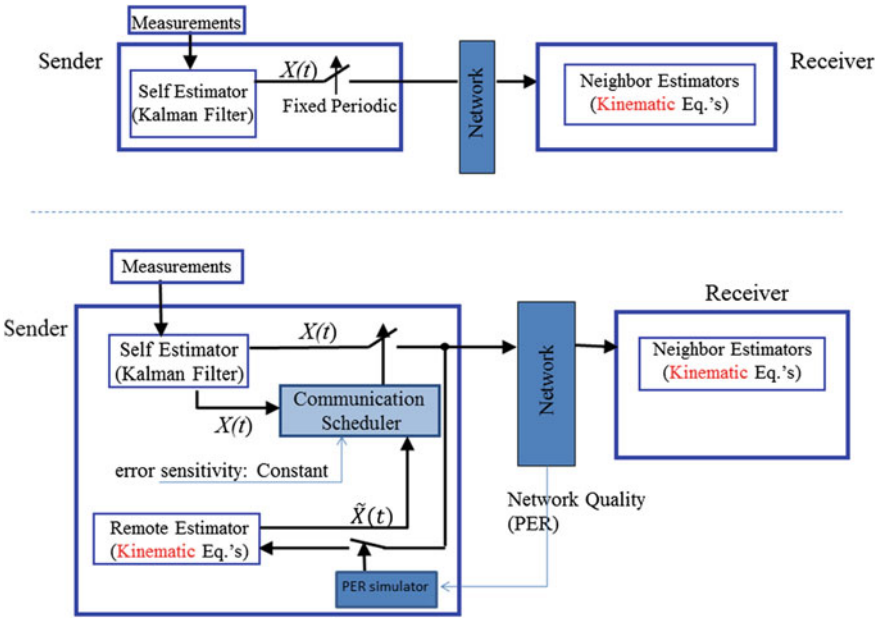


Fig. 8 Architecture of message generation algorithms. *Top* periodic message generation, *bottom* error-dependent message generation

The remote estimator is fed with the same messages that are transmitted over the wireless network to the receiver's "neighbor estimator." Since both estimators run the same model, their output will be the same if no messages are lost. Assuming no message loss, the output of the local "remote estimator" can be compared to the local state *X(t)* to determine the estimation error at the receiver (called suspected error). Packet loss increases the estimation error at the receiver. Since it is not possible to know exactly which messages have been lost (no acknowledgments in broadcast networks), an estimate of the packet loss rate will be used in the sender to adjust its approximation of the estimation error at the receiver.

The packet loss estimate is used as an input to a packet loss simulator module that randomly drops some of the messages that are sent to the remote estimator (see Fig. 8). The loss of some of the messages by this module will cause increased estimation error for the remote estimator [16], in a way simulating the increase in receiver's estimation error.

This method allows for close approximation of the estimation error at other nodes on an average basis but not for individual messages since exact pattern of received packets at each receivers is not known. As a result, the decision to transmit a specific sample *X(t),*measured at time *t,* has to rely on this approximate estimation error. To account for this uncertainty, we adopt a probabilistic method of deciding whether a message should be send or not. The idea is to send the message with a probability p that is directly related to the approximated estimation error. The higher the error, the more the probability of message transmission should be. This is achieved by defining the probability p as an increasing function of the approximated estimation error, $\hat{e}$, as follows (see Fig. 9):

$$p(t) = 1 - \exp(-\alpha.|\hat{e}(t) - e_{th}|^2) \text{ for } \hat{e}(t) > e_{th} \tag{3}$$

where $\hat{e}(t)$ is a node's approximation of the estimation error on neighboring vehicles (suspected error) toward its own position in a Euclidean sense (i.e., the usual distance definition for a Cartesian coordinate system), α is the error sensitivity, and e_{th} is the error threshold level. Below e_{th}, there is no need to send a message as approximated estimation error is acceptably low, so $p(t) = 0$.

Parameters α and e_{th} can be used to control the output rate of the message generator. Higher α and lower e_{th} will cause more messages to be generated. This design has been tested in a real-world implementation in [16]; however, it should generally be possible to use other functions for *p(t)*, or other methods for including the effect of packet loss, as long as they capture the same concept of increased message generation for higher approximated error.

The on-demand nature of Eq. (3) responds to the fact that a higher transmission rate, and thus probability, is required for a node that has more unexpected state change (as estimated by others). In information theoretical terms, a vehicle with higher entropy (a measure of surprise) needs a higher communication rate to describe its stochastic behavior.

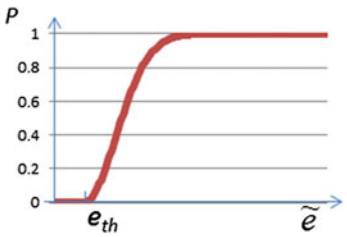


Fig. 9 The function relating $\hat{e}$ and probability of transmission p

On the contrary, when the suspected error is small or under the threshold, a node tends to stay quiet and not send a message, which will allow the channel to be used by those nodes that have larger suspected error (Fig. 9).

The use of Packet Error Rate (PER) to adjust the suspected estimation error at receiver nodes is an implicit way of correcting transmission errors. The reason is that PER estimate is used to increase the rate of message transmission to recover from lost messages. This is similar to retransmission in protocols like TCP, in which packet loss is detected through Acknowledgement messages and corrected through retransmission.

Employing PER for adjusting the suspected error of the receivers happens through a very simple simulation process. We use the measured PER to stochastically decide the suspected error $e_{th}(t)$ in Eq. 4: $\hat{e}(t+1) = (1-\eta)\hat{e}(t)$, where η is a Bernoulli trial (similar to coin tossing) with success probability 1-PER to address potential channel loss. If the trial is successful, the suspected error is reset; otherwise, suspected error accumulates from $\hat{e}(t)$ based on the known state model (e.g., Eq. 1). Note that this suspected error is not the actual estimation error at the receiver nodes; instead, it is only a measure used by a sender to adjust its own communication rate. Based on $p(t)$ in Eq. 3, a sender stochastically generates a message (i.e., containing the current sample of its state).

The PER used by the above Bernoulli trials is estimated on the fly by checking the inconsistency in sequence numbers of recently received packets from all corresponding senders (with at least two messages received) within a 1 second history log. That is, a node calculates the number of lost messages divided by the number of total messages sent by a certain sender to infer recent channel loss rate of that sender; PER is this measure averaged over all senders heard within a geographical area. Assuming network symmetry, PER tells a node the likelihood of the loss of its previous transmission at the receivers. Note that this PER is in fact an average and rough measure of the real error probability of individual links between the sender and receivers within the spatial neighborhood. The exact value of link error probability is not practically measurable under the assumptions of the broadcast networks, such as the one proposed for mobile sensor networks for active safety systems.

While Eq. (5) describes the probability of sending a new message at sampling intervals, the equivalent instantaneous rate of transmission that results from this probability can be described as follows:

$$R(t) = p(t)/T,\ p(t) = 1 - \exp(-\alpha \cdot |e(t) - e_{th}|^2)$$

It must be noted that the average rate produced by this policy should be measured over a sufficiently large number of samples (at least five) to get a correct sense of its effect on the network. The rate can be partly controlled through the tunable parameter α, which specifies how sensitive the controller is to the perceived error. A good value of this adjustable parameter is found to be around two and is not sensitive to the varying parameters of the network or dynamics of the tracked process [13, 14].

4.1 Rate-Distortion Modeling of the Message Generation and State Estimation Process

To design or analyze a rate control algorithm, it is often required that the relationship between the rate of transmission and the accuracy of estimation be found. This is similar to finding a rate-distortion model which tells us how rate and distortion (error or accuracy) are related. In certain cases, where the algorithms and wireless networks are simple, mathematical closed form solutions may be found [39, 40]. However, when complex algorithms such as the error-dependent policy are used with complex networks such as hidden node affected broadcast networks, closed form models become impossible or very difficult to derive.

In cases where direct mathematical derivation is not possible for rate-distortion modeling, empirical models are often used. This is in fact a routine practice in other domains like video where complex encoding and estimation methods are considered [26, 27]. In this section we use the same approach and characterize the relationship between the accuracy of tracking a real-time process (such as a mobile node position) and the rate of messages that are used. However, in order to exclude the effect of the network on the estimation accuracy, and to only evaluate the message generation and estimation process, we consider the rate of message "reception" instead of message transmission. This way, an estimation method can be evaluated in isolation, assuming an ideal network. The effect of network can then be modeled as a reduction in message rate, which can be applied to the overall model.

To find the empirical model, we consider two message generation policies of error-dependent and probabilistic (similar to and instead of periodic). We then observe how the rate of successful reception of messages (which is rate of transmission multiplied by success probability) affects tracking accuracy. The results are shown in Figs. 10 and 11 for a typical scenario (Highway speed 30 m/s, constant acceleration between samples, but acceleration changing to a new value according to a normal distribution N(0,1), creating an autoregressive process for speed). Here, we are using a range of message reception rates from 1 to 40 Hz and observing the resulting accuracy. Packet transmission is controlled by either randomly selecting whether the message should be transmitted (probabilistic policy), or using the error dependent policy that controls the rate based on perceived tracking error, according to [8] and similar to our method [13].

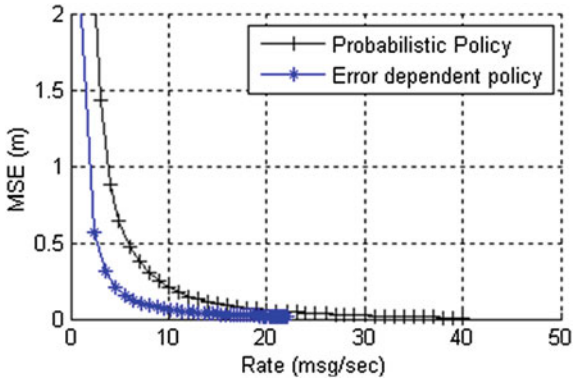


Fig. 10 Relationship between message arrival rate and tracking MSE

In Fig. 10 the MSE of tracking is shown, while in Fig. 11 a measure called 95 % cut-off error is depicted. The 95 % cut-off error is introduced as a statistical sense of the worst case behavior and is the value below which 95 % of the error histogram lies. We note that in wireless networks the worst case error is unbounded since there may be unbounded number of consecutive packet losses. Therefore a statistical measure like 95 % error must be used. The 95 % cut-off error (simply called 95 % tracking error) tells us that we can expect the estimation error to be less than this value, 95 % of the time.

To produce the results for different message rates (rate of reception in this experiment), the above algorithms were tuned with different ranges of parameters. For probabilistic policy, the rate is controlled by changing the probability of transmission at each sampling time (here set to 20 msec to examine higher rates, but normally set to 50 msec). For the error-dependent policy, the rate was adjusted through error-sensitivity parameter α (0.1–100), as described in Eq. 3. As expected, it is seen from these plots that higher rate of message arrival results in higher tracking accuracy (lower tracking error). However, the shape and position of the rate distortion curves are different for these algorithms, with error-dependent policy performing much better. This observation is consistent with well-known results [7, 8]. Moreover, we observe that the error saturates after a certain rate.

While these curves were generated for tracking a specific dynamical system, the general shape of the curves and the difference between the uncontrolled and controlled policies remain the same. Conceptually, with the same data rate, some policies are more "efficient" in the sense that they deliver information in a timely manner to the estimator to eliminate large tracking errors. The more the correlation of arrival information with the tracking error magnitude is, the lower the resulting tracking error will be.

Results shown in Figs. 10 and 11 point to the significant improvement that is possible using error-dependent policy, considering the fact that no communication overhead is needed and this policy has no extra system cost (e.g., coordination between nodes, etc.). Therefore, our choice for the message generation algorithm is the error-dependent policy. As it is observed from the rate distortion curve for this policy, the

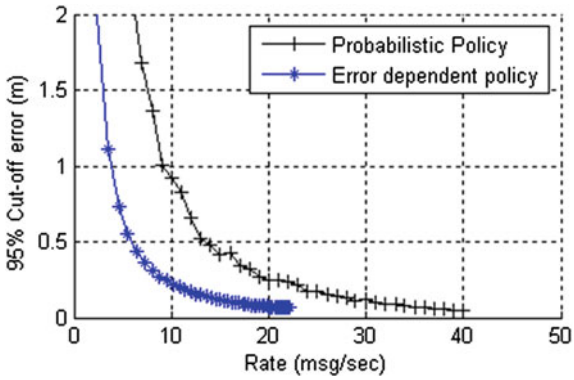


Fig. 11 Relationship between message arrival rate and 95 % tracking error

error drops quickly as message rate increases, but then saturates beyond a certain rate. It can be deduced that the estimation performance does not improve after a certain rate, at which the error is already low. At this point, it might be more useful to use network resources to reach farther nodes, than to send more packets to closer nodes. This fact is used in the design of the overall transmission control protocol by controlling the rate and maintaining it above an acceptable level to produce accurate tracking, and controlling the range to cover more nodes while avoiding congestion. The next section describes the range control approach.

5 Controlled Policy for Transmission Range Adaptation

Following the discussion on transmission rate adaptation, and considering the fact that rate and range can be controlled separately to maximize IDR, the objective of this section is to introduce methods for maximizing tracking range without sacrificing tracking accuracy. Achieving this objective will be possible if we can ensure that through some algorithms (like the error-dependent policy) nodes are maintaining high enough rates of message reception, followed by a mechanism that increases tracking range. From a network measure perspective, this is like increasing the range as long as IDR is improving. Improvement in IDR may come as a result of more nodes receiving the messages.

If error-dependent rate control was not used, with increase in range, reception rate may have dropped while IDR (seen in the form of product of reception rate and number of receivers) might still increase. However, the rate control scheme ensures that reception rate stays high and fixed by increasing the transmission rate. With almost fixed reception rate, increase in IDR means increase in the number of receivers. Therefore, the policy of choice could be to maximize IDR under the assumption of controlled transmission rate (fixed reception rate).

Figure 5 shows that for any value of transmission rate, there is a value for range that maximizes IDR. To see this better, assume a fixed transmission rate (not reception

rate), when IDR is below the maximum point, increase in range will cause more nodes to receive the messages and IDR improves. Beyond the maximum IDR point, increase in range causes faster drops in reception rate than increase in the number of receivers. Around the maximum point, increase in range is met with decrease in reception rate, keeping IDR almost fixed. This means that if no rate control scheme is used, IDR should not go near and beyond its maximum, and possibly the operation point should be at points before maximum IDR happens. With a rate control scheme that tries to maintain the rate of reception, when increase in range and IDR causes lower reception rate, the algorithm pushes toward an operation point that has higher transmission rates. From Fig. 5 this is like moving a plane cut on the IDR surface, which describes IDR versus d for a given r, towards higher values of r; the maximum IDR in this case will be achieved at a lower range d. The reduced range ensures that the algorithm is never operating above the maximum point where reception rate is sacrificed for better range.

Adapting the range of transmission requires that some feedback is available on how the network is performing. In [29], an implicit method of controlling the load on the network through broadcast of the selected power values was proposed; however, the presented method did not consider the actual network performance and how it relates to the presented load. The algorithm to measure the load also required cooperation of all the nodes in a neighborhood. The works in [10, 14] take a different approach and try to localize the action as much as possible and achieve range control with least coordination between nodes. The methods in [10, 14] directly consider how the network performs by local measurement of readily available channel feedback measures such as channel busy ratio, and by relating this feedback measure to network performance IDR.

The idea is inspired by the fact that TCP achieves congestion management in the Internet with almost no coordination of actions between end nodes. Each node in TCP monitors the locally observable round trip time and adjusts the rate of transmission in a way that avoids congestion in the routers on the path of TCP flows. A readily available, locally observable, feedback measure from the wireless network is "channel occupancy" or "channel busy ratio (CBR)." We had introduced the use of this measure as an appropriate feedback measure in [13, 15] and later presented a detailed analysis in [11]. This feedback measure was available in some 802.11p devices around that time and is now universally available on all common 802.11p devices. Recent works by other researchers have also taken advantage of the CBR measure [38]. To understand how CBR can be used, the next subsection describes the relationship of CBR and network performance.

5.1 Channel Busy Ratio as Feedback Measure

To understand how channel occupancy (busy ratio) is related to the network performance measure, IDR, we use simulation experiments [12] and analytical results from [11]. Channel busy ratio, denoted as u, is measured by computing the ratio of

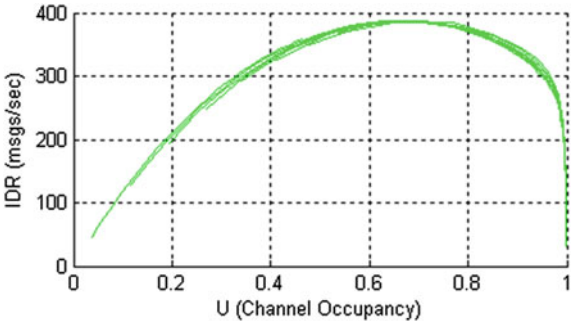


Fig. 12 IDR versus channel occupancy for different values of r (5–115 msg/sec), d (20–400 m), and ρ (0.1–0.8 vehicle/m, considering an eight lane highway). Points belonging to the same experiment with different values of d are connected by a line; although due to overlap they are indistinguishable. [11]

time that the channel is sensed busy in a given window of time T (in the order of hundreds of milliseconds to seconds). If channel status at each minislot of duration T_{slot} is measured, u is found as follows:

$$u = \left(\sum_{i=1}^{[T/T_{slot}]} \Lambda_i \right) / [T/T_{slot}] \tag{4}$$

where Λ is 1 for busy minislots and 0 otherwise, and $[T/T_{slot}]$ is the number of minislots in the measurement window. In practice the measurements are done through observing results from 802.11 Clear Channel Assessment (CCA) functionality and measuring durations of idle and busy period.

In Fig. 12, *IDR* is plotted against channel busy ratio, for a large range of values of r, d, and road densities ρ. Each choice of (r, d, ρ) generates one (*u, IDR*) point. Interestingly, it is observed that the resulting (*u, IDR*) points, for all different choices of parameters, fall on a single dome shaped curve (thus not separable in the figure). It can be deuced that the relationship between *IDR* and u is not a function of the parameters r, d, or ρ. It was reported in [11] that the shape is a result of the type of protocol (CSMA/CA, Aloha, etc.) and the type of interference (pure CSMA/CA or hidden node affected CSMA/CA), as well as MAC and PHY parameters. While the result is interesting, it is not totally unexpected. In simple terms, *IDR* represents the total channel capacity, and channel busy ratio is related to the offered load. For simple protocols such as ALOHA it is possible to even mathematically derive the relationship between *IDR* and u in a closed form.

To further see how different protocols and situations affect *IDR*, in Fig. 13 *IDR* is plotted for a CSMA network with no hidden node effect, for a CSMA/CA vehicular network with hidden node effect but no PHY layer, and for CSMA/CA vehicular network with hidden node effect and PHY layer. Figure 13 shows the "normalized *IDR*" in order to provide further insight into the effect of different protocols. Normal-

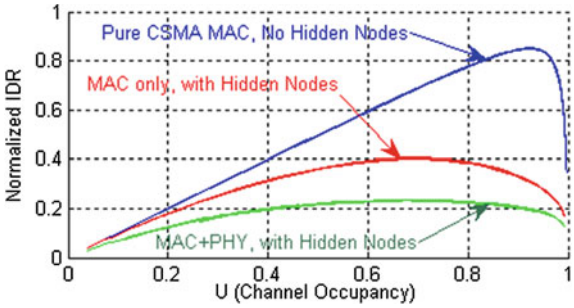


Fig. 13 Normalized IDR versus channel occupancy for different settings. [11]

ization is done with respect to the full capacity of the channel if access was perfectly and ideally scheduled amongst all nodes ($IDR_{sch} = 1/T_p$ packets per second).

The fact that the relationship of *IDR* versus u can be described regardless of the choice of (r, d, ρ) is observed for all of the above scenarios. It is also observed that the presence of hidden node interference causes the total *IDR* to be much lower than a fully connected CSMA network. The channel busy ratio at which maximum IDR is observed is different for each situation, and is at lower values for the case of hidden node interference. Therefore, it is concluded that the use of channel busy ratio to estimate *IDR* should consider the protocol and interference model (hidden nodes in our case).

Considering the definition of *IDR* as the number of messages delivered to all neighbors in a given time period, we can describe *IDR* measurement as averaging reception rates over all distances in the neighborhood. Similarly, one could also think of a weighted average. This way messages that are delivered to different distances get a different weight; close nodes are valued more than far nodes. In Fig. 14 (top plot), we demonstrate the weighted *IDR* (wIDR) for 3 weighting functions: (1) linear (2) quadratic (3) constant [11]. The constant weight produces the original *IDR*.

In Fig. 14, different *wIDR* versus u curves for range values of 50–600 m are plotted. To better visualize, $(u, wIDR)$ points are plotted as a connected curve for all the values range 50–600 m; different curves are plotted for each value of rate. The rate was varied from 5–150 msg/sec with step size of 10.

It is observed that for original IDR, all curves of (*u, IDR*) points fall on a single dome shaped curve and overlap; for the other options the connected curves do not completely overlap. For higher rates $r > 15$, *wIDR* points are also very closely located. For very small rates such as $r = 5$, the curve is far from the set of other curves. In general, it is observed that with weight functions that have higher weight at closer nodes (e.g., linear decreasing function), the channel busy ratio at which maximum *wIDR* happens is spread over a larger set of values, which makes optimized designs more dependent on the transmission rate. Note that the absolute values of different *wIDR* options should not be compared to each other in this figure.

The choice of which weighting function to use is outside the scope of this paper, as it is an application specific choice. Nevertheless, we present an example of a

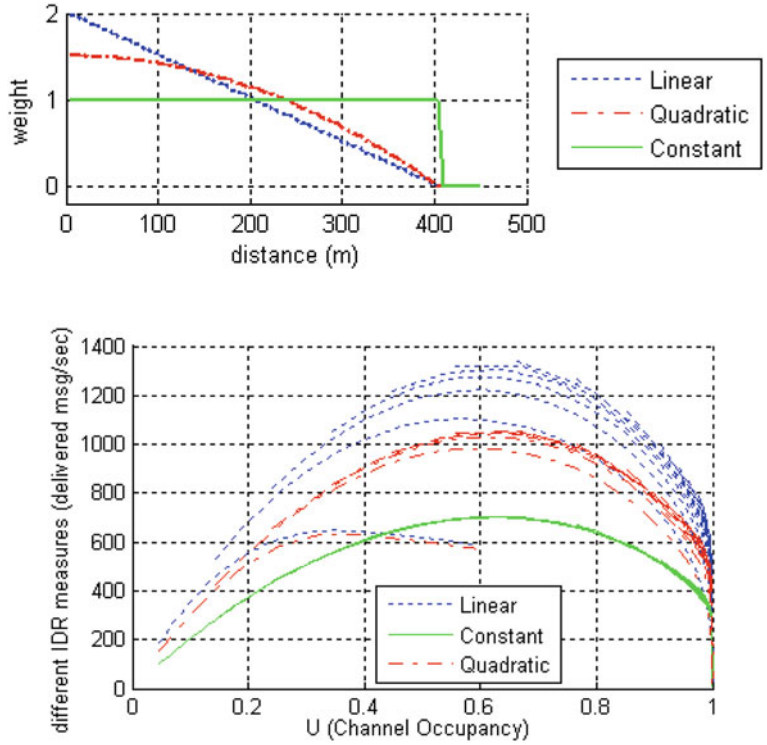


Fig. 14 Different choices of weighting function (*top*), weighted IDR versus Channel Occupancy (U) for different weighting function (*bottom*). PHY effect is not shown to accentuate the difference of curves, otherwise results are similar to the above [11]

design based on one of these choices (normal IDR) in the next section, to provide a comprehensive guide for designing congestion management protocols.

The observation that IDR and channel busy ratio are related regardless of the controllable parameters r and d motivates the use of channel busy ratio as a feedback measure to control the network to achieve maximal IDR. For example, one could devise a feedback control scheme that adapts the range or rate of transmission to maintain the observed channel busy ratio around its optimal point (0.7 for highway setting of a vehicular network), where maximal information dissemination happens.

5.2 Robust Transmission Range Control Based on CBR

From the relationship between *IDR* and channel occupancy (channel busy ratio), it is induced that a feedback control scheme can be designed to adjust the rate and range of transmission so that channel occupancy, hence IDR, is maintained near its optimal

value. Different designs based on the above concept have been proposed in our previous works [10, 14]. The idea in these designs is that the range of transmission can be adapted to network conditions, assuming that the rate of transmission is dictated by the requirement of the estimation process ([13, 14]) and is controlled to maintain reception rate above a fixed "good" value. The range control is therefore free to explore different methods of achieving maximum IDR.

To see how transmission range d can be controlled to operate near optimal channel busy ratio, we first study how the range of transmission (controllable parameter) is related to channel busy ratio (the observed measure). For this purpose we plot the relationship between channel busy ratio and transmission range d, for different choices of r and node density ρ in Fig. 15. As expected channel occupancy increases with increase in range or rate. Different values of rate or traffic density lead to different curves describing u versus d.

The u versus d curves can be seen as plant models, describing the network behavior for given parameters r and ρ. The general control idea is to observe channel busy ratio u and throttle range d if u increases. However, this should be done in a way that channel busy ratio is near its optimal value. Denoting the relationship between d and u by function $u = h_r, \rho(d)$, many different decreasing functions of u (i.e., $d = g(u)$) can be designed to achieve this behavior. Such range (power) control measures will over time, under certain stability conditions, result in a solution that sits at the intersection point of $u = h_r, \rho(d)$ and $d = g(u)$. However, since each rate r or density ρ generates a different $u = h_r, \rho(d)$ curve, the intersection point will be different for each r or ρ. This means that the feedback control scheme converges to a different point for each different r or ρ. Given that the values of r and ρ are dependent on many factors including node movement, it is not practical to know a priori which one of the $h_{r,\rho}(d)$ curves characterizes the network for any given case. Therefore, we resort to a robust, but sub-optimal design of controlling the value of d in a way that channel busy ratio stays in a *range* of interest. The range of interest can be easily determined from Fig. 12, where we can see that the value of u between 0.4 and 0.8 results in near-peak IDR (for further stability, in real implementations 0.35 and 0.85 are used). As it was mentioned earlier, the right side of the IDR maximum point is not desirable since it infers large packet loss and a very busy medium leaving no space for other applications; therefore, we try to avoid CBR of near 0.9. The following control function can achieve the objective of maintaining *CBR* in the desired range:

$$d = g(U) = \begin{cases} D_{max} & u < U_{min} \\ D_{min} + \frac{U_{max} - u}{U_{max} - U_{min}}(D_{max} - D_{min}) & U_{min} \leq u < U_{max} \\ D_{min} & U_{max} \leq u \end{cases} \quad (5)$$

where D_{min} and D_{max} denote the minimum and maximum allowed ranges determined by the application and practical constraints, and U_{min} and U_{max} are determined from Figs. 12 or 14 (0.35 and 0.85). We call this controller the Linear Range Control or LRC. The LRC control function and its relation to $u = h_r, \rho(d)$ is shown in Fig. 15.

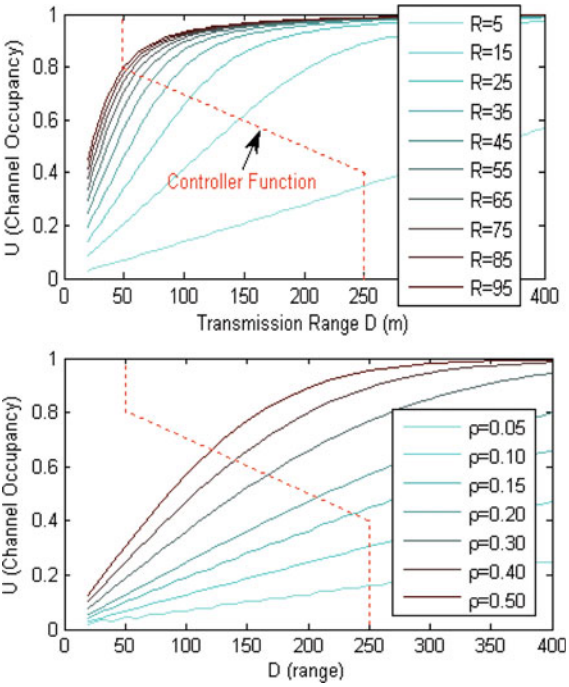


Fig. 15 Channel busy ratio $u(U)$ versus range d (D), for (*top plot*) ρ=0.2 and different transmission rates: $u = h_R(d)$, or (*bottom plot*) $R = 10$ and different density $\rho : u = h_\rho(d)$; sample range control function $d = g(u)$ is also shown [11]

Another option for adapting range of transmission to achieve optimal CBR is the Gradient descent Range Control (GRC) algorithm of [10], which tries to keep CBR at its optimal value (e.g., $u^* = 0.7$) for IDR maximization. The algorithm uses gradient descent update to the range value in each iteration:

$$d_{k+1} = d_k + \eta(u* - u_k) \tag{6}$$

In the above equation, d_k denotes the value for range at kth update, and u_k is the resulting measured channel busy ratio. The time interval between updates is T, and u is calculated as in (4). Clearly, when the algorithm converges the value of d will be such that $u = u^*$ and maximum IDR is achieved. However, for the above algorithm to converge, the value of η has to be selected in a way that: (1) d converges quickly to near optimal value, before the value of ρ or average value of r changes considerably (2) the system does not overshoot too much or oscillate and stray into a region that yields significantly low value of IDR (e.g. $u > 0.95$ or $u < 0.3$ in Fig. 12). A convergence study of the GRC algorithm has been reported in [10] where appropriate range of η has been found for a feedback linearized version of GRC with a modified update equation:

$$d_{k+1} = d_k + \eta \ln(\frac{1 - u_k}{1 - u*}) \tag{7}$$

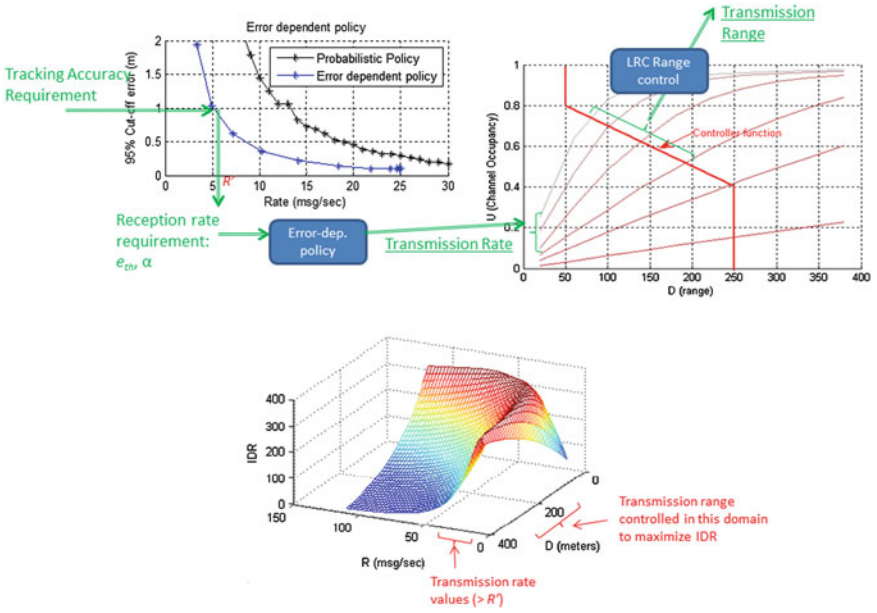


Fig. 16 The design flow and controller operation ranges; the reverse flow of PER and CBR is not shown

While GRC is generally more appropriate and has less stability issues than LRC (since its range of stable gains can be found mathematically), it has the drawback of relying on a single value of u^*. The issue is that this value has to be found for each network situation and may slightly vary for different cases; however, a value like what is seen in Fig. 12 ($u^* = 0.7$) will be close enough to the optimal value for different choices of parameters for the CSMA/CA protocol. Change in network topology requires a different analysis though; the results in Fig. 12 are for a highway-like linear topology.

Given the above designs for range control, and the rate control algorithm of previous section, we are now able to revisit the design of the transmission control protocol and describe the overall design in the next section.

5.3 Protocol Summary and Evaluation

The designs presented in previous sections for adapting transmission rate and range can be seen together as a transmission control protocol to support real-time tracking over uncoordinated wireless networks. The combined protocol provides a TCP-like mechanism that adjusts application's "load" in order to serve its objective. Here the objective is real-time tracking, but for TCP the objective is to achieve sustained high

throughput. The load in TCP is identified as the rate of transmission, whereas the load in the case of a spatially distributed wireless network can be defined in terms of the time and space resource that is taken by each node [44]. Controlling access to the time and space resource translates to rate and range control.

The combined algorithm can be explained in a two-step design flow:

- First the desired accuracy of tracking is decided and then used in the rate-distortion model to derive the required rate of message reception. The transmission rate control algorithm, an error-dependent policy, is then responsible for maintaining the reception rate at this level by adjusting the transmission rate according to the process dynamics and a PER estimate from the network.
- Second, the range of transmission is controlled according to LRC or GRC, based on observed CBR, in order to maintain IDR at high levels, resulting in increased range while keeping the rate above the required levels.

The above design is illustrated in Fig. 16.

CBR is directly affected by the transmission rate, which is in turn due to the error dependent policy and the density of nodes. A higher CBR (beyond u^*) may lead to a higher PER value; the result is higher transmission rate which further increases CBR if no range control was employed. However, the range control algorithm is much more aggressive in reducing the range, than the rate controller is in increasing the rate. The speed of rate increase becomes slower at higher rates (saturates) as is seen in Fig. 9, while the LRC range control method maintains its gain over all possible range values. The result is that the possible positive feedback that is observed because higher rates may cause higher PER and in turn even higher rates, is aggressively offset by the range control scheme that brings the range and PER down very quickly. The reduction in PER will be followed by a reduction in transmission rate and therefore CBR. At this point the reduced CBR will allow the range control to move to a higher value and the process continues until a balance is reached.

5.3.1 Evaluation

To see how the transmission protocol performs in practical settings, a relatively realistic simulation experiment was conducted. In this experiment, the mobile sensor network was an ad hoc network of vehicles broadcasting their state information (position, speed, etc.) to their neighborhood. Node movements were derived using a microscopic traffic simulator called SHIFT [41], simulating a 2 km span of a highway with different traffic density scenarios (Table 1). The highway was assumed to have four lanes of identical traffic in each direction. Traffic in different directions may have different density. We only collect statistics from vehicles within a 0.5–1.5 km segment to avoid boundary effects in simulations. The simulation scenarios in Table 1 include two sets of homogeneous and mixed traffic for the two directions. Cases H1–H4 represent scenarios with similar (homogeneous) traffic flows in both directions, while cases M1–M6 contain mixed traffic flow conditions for the two directions. Mean flow

Table 1 Simulated bidirectional highway traffic scenarios [14]

Case ID	Direction #1 Status	Direction #1 Speed (mph)	Direction #2 Status	Direction #2 Speed (mph)
H1	Congested	14	congested	14
H2	Low speed	30	Low speed	30
H3	Medium speed	53	Medium speed	53
H4	Free flow	74	Free flow	74
M1	Congested	14	Low speed	30
M2	Congested	14	Medium speed	53
M3	Congested	14	Free flow	74
M4	Low speed	30	Medium speed	53
M5	Low speed	30	Free flow	74
M6	Medium speed	53	Free flow	74

speed, in the unit of miles per hour, is also listed for reference. These values and designations are typical in transportation research.

The trajectory of vehicles from SHIFT, sampled at 20 Hz for 30 s, is fed to OPNET network simulator. We implemented our algorithms over a modified 802.11a network (to simulate DSRC) in OPNET. The modified 802.11a PHY module working at 5.9 GHz with 10 MHz bandwidth resembles DSRC. We follow the DSRC channel model reported in [28] and simplify the far distances as Rayleigh fading instead of pre-Rayleigh. The rationale for this simplification is that we are considering a straight highway scenario, while [28] considers urban scenarios with intersections and corners, which lead to pre-Rayleigh fading observations. In these simulations, the path loss exponent was set to 2.31. The transceiver was assumed to operate with –87 dBm receiver sensitivity, at 3 Mb/s raw data rate. For the 100 and 500 ms periodic beaconing schemes, a fixed 28 dBm transmission power was used (which roughly covers the radius of 250 m). This power value is suggested by VSCC [5]. The payload size of each message is 300 bytes.

Each vehicle's communication module in the experiment samples its state at 50 msec time steps. The communication logic, following the specified rate control scheme, then decides whether or not to generate a packet. The onboard measurement noise is modeled exactly the same way as in [6], which was in turn based on experiment data [1].

Upon receiving information from the shared channel, each vehicle updates its estimation of the sending car's position. Vehicles maintain a map of all cars in their proximity. A simple first-order kinematic model (i.e., a constant speed predictor) is used for tracking neighboring vehicles. Using this predictor, a vehicle is assumed to run at the same speed and in the same direction after its last successful information broadcast is received by the predictor.

5.4 Discussion

In each simulation run, data such as the specific packets that are sent or received by nodes are collected. This information is used in calculating how the estimation process in each node progresses and what is the estimation error of each node, of the position of other nodes, at any given time. The results are then processed to derive statistics such as 95 % tracking error (calculated from **Euclidean tracking error)** over all vehicles in predefined distance interval bins from any sender to explore the law of large numbers. The bins of distance are shown in Fig. 17 (x-axis of all the plots). We use the 95 % cutoff error as the main performance metric for comparisons in Fig. 17. The main advantage of this performance measure over others (e.g., mean or **standard** deviation of error) is that it gives a statistical sense similar to that of a confidence interval.

To see how the proposed transmission control protocol performs, we compare its tracking accuracy with that of uncontrolled periodic beaconing methods (at 100 and 500 ms intervals) for the scenarios in Table. 1. The following parameters are used for the transmission control protocol: $\alpha = 2\,\mathrm{m}^{-2}$, $e_{th} = 0.2\,\mathrm{m}$, $L_{\min} = 50\,\mathrm{m}$, $L_{\max} = 250\,\mathrm{m}$, $U_{\min} = 0.4$, and $U_{\max} = 0.8$.

The tracking accuracy (95 percent cutoff Euclidean error) for both 100 and 500 ms beaconing, and our proposed solution is plotted in Fig. 17 with respect to various traffic conditions in Table 1. The result in each subplot shows the tracking accuracy of the above communication policies at different distance intervals. distance intervals are presented in 30 m bins between 0–240 m. The reason for such presentation and averaging the results in each bin is to produce better statistical results. The results illustrate how tracking accuracy degrades as the distance from a sender increases. Only results from cases H1, H2, M3, M5, and M6 are presented in Fig. 17. For cases H1 and H2, only tracking accuracy in direction #1 is presented since traffic in both directions is homogeneous; for cases M3, M5, and M6, tracking accuracy in both direction #1 (left) and direction #2 (right) is presented. For the other scenarios the results of the periodic beaconing (100 ms interval) and the proposed transmission protocol are similar, since there is no strain on the network and no congestion issues.

The message rate for 100 ms beaconing is 10 packets/s, while it is 2 packets/s for 500 ms beaconing. Our proposed design produces a message rate of 2–3 packets/s for the cases shown in Fig. 17. The rate is dependent on node movement and network condition, so it is variable. On average, the proposed transmission control protocol achieves better (or equal) tracking accuracy than beaconing methods. As observed from Fig. 17, the 100 ms beaconing method does not scale well when facing different traffic conditions, especially when there is a high density of cars. This uncontrolled method suffers from consecutive message losses which result in higher tracking error.

The beaconing method with message rate of 2 Hz (500 ms interval) does not suffer channel congestion in most cases because its transmission load is much lower than the 100 ms beaconing. Therefore, in some cases 500 ms beaconing achieves better tracking performance than 100 ms beaconing due to lower packet collisions and loss. However, its large inter-message interval is too long for some cases where node

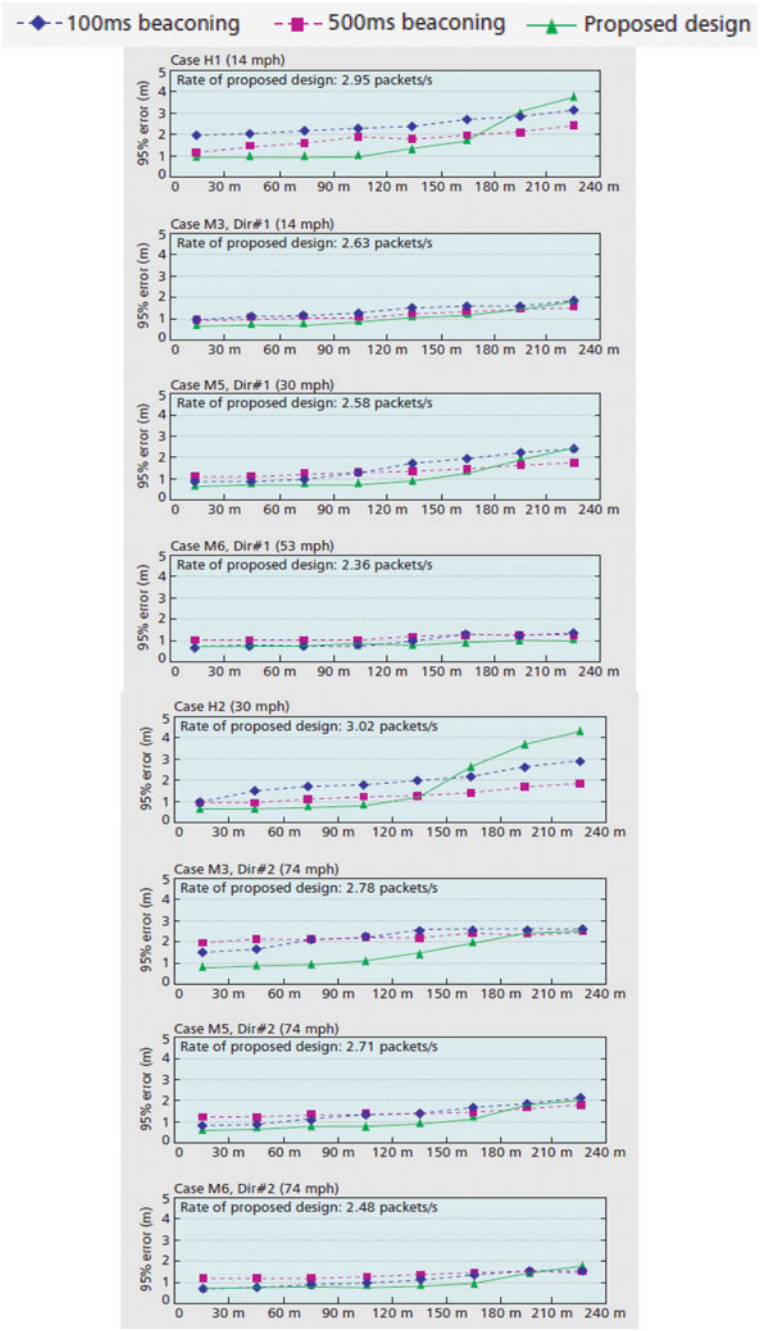


Fig. 17 Tracking accuracy at different distances from a sender

movement is high; in such cases, not enough samples are available for neighboring vehicles to track a sending node in real-time. As a result, in most scenarios the tracking error of 500 ms beaconing is still larger than where our transmission control protocol is used. Although the channel congestion and load level produced by the proposed transmission control protocol is similar to that of 500 ms periodic beaconing, the proposed method is more efficient and uses timely transmission of state information to eliminate large tracking errors.

Furthermore, the range control component of the transmission control protocol adapts range (power) to maintain an acceptable level of channel occupancy. This means that when the network is crowded a lower range will be used. The result is that tracking error stays flat and low for a certain range or distance (this range depends on vehicular traffic condition and network congestion) and increases beyond the range of interest. This is in fact a desired effect since in crowded networks there are many vehicles in close proximity of a given car, and it is more important to track the close vehicles more accurately than to track both close and far vehicles with less accuracy. Therefore, the range control method gives up on far nodes, when there are many vehicles nearby. This is seen in the results as tracking error of the farther nodes increase beyond the range of interest. Take the H1 case (Fig. 17), for example; the 95 % Euclidean tracking error stays roughly the same within a 90 m radius; beyond that, the tracking error goes up quickly as distance increases. In other words, the immediate neighbors have better tracking accuracy than others. Here, the algorithm has responded to congestion by trying to maintain the same amount of state information available to its closer neighbors, sacrificing the far nodes which are not actually important from a safety and tracking perspective.

6 Summary

This chapter discussed real-time tracking over a wireless network, in particular in uncoordinated networks such as those found in wireless and mobile sensor networks. It was shown that controlled policies for transmission of state information can significantly improve the performance of tracking applications. In particular if transmission time, frequency, and range is tied to the dynamics and requirements of the process that is being tracked over the network. To this end, a transmission control protocol that operates in rate and range domains (time and space) was presented and its analogy to TCP discussed. The transmission control protocol, and in particular its range control component, has been designed so that the effect of the real-time tracking application on the uncoordinated wireless network is not detrimental to the application itself. The uncoordinated networks (like ad hoc networks of DSRC) suffer from significant loss of capacity if their load increases beyond certain limits, requiring congestion management and transmission control protocols at the end nodes.

References

1. R. Sengupta, S. Rezaei, S.E. Shladover, J.A. Misener, S. Dickey, H. Krishnan, Cooperative collision warning systems: concept definition and experimental implementation. J. Intell. Transp. Syst. **11**(3), 143–155 (2007)
2. IEEE 802.11 WG, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, IEEE, Aug. 1999
3. Wireless Access in Vehicular Environment (WAVE) in Standard 802.11, Specific Requirements: IEEE 802.11p/D1.0, Feb. 2006
4. IEEE 1609.3 Wireless Access in Vehicular Environment: Networking Services, 2007
5. Vehicle Safety Communications Consortium (VSCC), Vehicle Safety Communications Project, Task 3 Final Report: Identify Intelligent Vehicle Safety Applications Enabled by DSRC, 2005
6. S. Rezaei, R. Sengupta, H. Krishnan, X. Guan, R. Bhatia, Tracking the position of neighboring vehicles using wireless communications. To appear in Elsevier Journal Transportation Research Part C: Emerging Technologies, Special Issue on Vehicular Communication Networks
7. B. Sinopoli, et al., Kalman filtering with intermittent observations. IEEE Trans. Automat. Contr. **49**(9) (2004)
8. Y. Xu, J. Hespanha, Communication logics for networked control systems, in *Proceedings of the American Control Conference*, June 2004
9. Y.P. Fallah, F. Agharebparast, M. Minhas, H.M. Alnuweiri, V.C.M. Leung, Analytical modeling of contention-based bandwidth request mechanism in IEEE 802.16 wireless networks. IEEE Trans. Veh. Tech. **57**(5), 3094–3107 (2008)
10. Y.P. Fallah, C.L. Huang, R. Sengupta, H. Krishnan, Congestion control based on channel occupancy in vehicular broadcast networks, in *Proceedings of IEEE Vehicular Technology*, Conf VTC Fall, 2010
11. Y.P. Fallah, C.L. Huang, R. Sengupta, H. Krishnan, Analysis of information dissemination in vehicular AdHoc networks of cooperative vehicle safety systems. IEEE Trans. Veh. Technol. **60**(1), 233–247 (2011)
12. Y.P. Fallah, C.L. Huang, R. Sengupta, H. Krishnan, Design of cooperative vehicle safety systems based on tight coupling of communication, computing and physical vehicle dynamics, in *Proceedings of International Conference on Cyber Physical Systems*, April 2010
13. C. L. Huang, Y. P. Fallah, R. Sengupta, H. Krishnan, Information dissemination control for cooperative active safety applications in vehicular Ad-Hoc networks, in *Proceedings of IEEE Globecom* 2009
14. CL. Huang, Y. P. Fallah, R. Sengupta, H. Krishnan, Adaptive inter-vehicle communication control for cooperative safety systems. IEEE Netw. **24**(1), 6–13 (2010)
15. C. L. Huang, X. Guan, Y. P. Fallah, R. Sengupta, H. Krishnan, "Robustness Evaluation of Decentralized Self-Information Dissemination Control Algorithms for VANET Tracking Applications" Proc. of IEEE Vehicular Technology Conf. Fall-2009.
16. C.L. Huang, H. Krishnan, R. Sengupta, Y.P. Fallah, Implementation and evaluation of scalable vehicle-to-vehicle safety communication control. IEEE Commun. **49**(11), 134–141 (2011)
17. C.L. Huang, Y. P. Fallah, R. Sengupta, H. Krishnan, Inter-vehicle transmission rate control for cooperative active safety system. IEEE Trans. Intell. Transp. Syst. **12**(3), 645–658 (2011)
18. R. Caceres, L. Iftode, Improving the performance of reliable transport protocols in mobile computing environments. IEEE J. Sel. Areas Commun. **13**(5) (1995)
19. H. Balakrishnan, V.N. Padmanabhan, S. Seshan, R.H. Katz, A comparison of mechanisms for improving TCP performance over wireless links. IEEE/ACM Trans. Netw. **5**(6), 756–769 (1997)
20. Chonggang Wang, Mahmoud Daneshmand, Bo Li, Kazem Sohraby, A survey of transport protocols for wireless sensor networks. IEEE Netw. **20**(3), 34–40 (2006)
21. Y. G. Iyer, S. Gandham, S. Venkatesan, STCP: a generic transport layer protocol for wireless sensor networks, in *Proceedings of IEEE ICCCN 2005*, San Diego, California, Oct. 17–19
22. B. Hull, K. Jamieson, H. Balakrishnan, "Mitigating congestion in wireless sensor networks, " in Proceedings of ACM Sensys'04, Nov. 3–5, Baltimore (Maryland, USA, 2004).

23. C.-Y. Wan, S.B. Eisenman, A.T. Campbell, CODA: congestion detection and avoidance in sensor networks, in *Proceedings of ACM Sensys'03*, Los Angeles, California, Nov. 5–7 2003
24. C.-T. Ee, R. Bajcsy, Congestion control and fairness for many-to-one routing in sensor networks, in *Proceedings of ACM Sensys'04*, Baltimore, Maryland, Nov. 3–5 2004
25. C. Wang, K. Sohraby, V. Lawrence, B. Li, Priority-based congestion control in wireless sensor networks, Accepted to appear in The *IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing* (*SUTC2006*), Taichung, Taiwan, June 5–7 2006
26. H. Mansour, Y.P. Fallah, P. Nasiopoulos, V. Krishnamurthy, Dynamic resource allocation for MGS/AVC H.264 video transmission over link-adaptive networks. IEEE Trans. Multimedia **11**(8), 1478–1491 (2009)
27. Y.P. Fallah, H. Mansour, S. Khan, P. Nasiopoulos, H. Alnuweiri, A link adaptation technique for efficient transmission of H.264 scalable video over multirate WLANs. IEEE Trans. Circ. Syst. Video Technol. **18**(7), 875–887 (July 2008)
28. L. Cheng, B. E. Henty, D. D. Stancil, F. Bai, P. Mudalige, Mobile vehicle-to-vehicle narrow-band channel measurement and characterization of the 5.9 GHz dedicated short range communication frequency band. IEEE J. Sel. Areas Commun. **25**(8) (2007)
29. M. Torrent-Moreno, J. Mittag, P. Santi, H. Hartenstein, Vehicle-to-vehicle communication: fair transmit power control for safety-critical information. IEEE Trans. Veh. Technol. **58**(7) 3684–3703 (2009)
30. C. Robinson, L. Caminiti, D. Caveney, K. Laberteaux, Efficient coordination and transmission of data for vehicular safety applications, in *Proceedings of the 3rd ACM VANET*, Sept. 2006
31. S. Rezaei, R. Sengupta, H. Krishnan, X. Guan, Reducing the communication required by DSRC-based vehicle safety systems, in *Proceedings of 4th ACM VANET*, Sept. 2007
32. Y. Xu, J. Hespanha, Estimation under uncontrolled and controlled communication in networked control systems, in *Proceedings of Conference on Decision and Control*, Dec. 2005
33. M. Torrent-Moreno, J. Mittag, P. Santi, H. Hartenstein, Vehicle-to-vehicle communication: fair transmit power control for safety-critical information. to appear in IEEE Trans. Veh. Technol.
34. S. Ray, D.Starobinski, J. Carruthers, Performance of wireless networks with hidden nodes: a queueing theoretic analysis. J. Comp. Commun. (SI on Perf. Issues of WLANs, PANs, and Ad Hoc Networks) **28**(10), 1179–1192
35. J. Yin, T. Elbatt, G. Yeung, B. Ryu, S. Habermas, H. Krishnan, T. Talty, Performance evaluation of safety applications over DSRC vehicular Ad Hoc networks, in *Proceedings of 1st ACM VANET*, Sept. 2004
36. J. Mittag, F. Schmidt-Eisenlohr, M. Killat, J. Harri, H. Hartenstein, Analysis and design of effective and low-overhead transmission power control for VANETs, in *Proceedings of 5th ACM VANET*, Sept. 2008
37. P. Seiler, R. Sengupta, An H ∞ approach to networked control. IEEE Trans. Autom. Control **50**(3), 356–364 (2005)
38. Q. Chen, D. Jiang, T. Tielert, L. Delgrossi, Mathematical modeling of channel load in vehicle safety communications, in *Proceedings Of IEEE Wireless Vehicular Communication Symposium*, 2011
39. C.L. Huang, R. Sengupta, Analysis of channel access schemes for model-based estimation over multiaccess networks, in *Proceedings of IEEE MSC*, Sep. 2008, pp. 408–413
40. C. L. Huang, R. Sengupta, Decentralized error-dependent transmission control for estimation over a multiaccess network, in *Proceedings of 4th WICON*, Nov. 2008, p. 80
41. SHIFT: California PATH SHIFT. [Online: http://path.berkeley.edu/SHIFT/]
42. OPNET Modeler 14.0. [Online]. Available: http://www.opnet.com/
43. V. Jacobson, Congestion avoidance and control, in *Symposium Proceedings on Communications Architectures and Protocols* (*SIGCOMM '88*), ACM, pp. 314–329
44. P. Gupta, P.R. Kumar, The capacity of wireless networks. IEEE Trans. Info. Theory **46**(2), 388–404 (2000)

Chapter 7
Target Counting in Wireless Sensor Networks

Dengyuan Wu, Bowu Zhang, Hongjuan Li and Xiuzhen Cheng

Abstract Target counting in wireless sensor networks has attracted a lot of attention in recent years from both academia and industry. In this chapter, we review various problem formulations and technical approaches proposed in recent literature for target counting. Major existing works are classified into the following four categories: binary counting, numeric counting, energy counting, and compressive counting, based on the sensing capabilities of the network and the underlying theoretical foundations of the technical approaches. Within each category, we summarize the representative works according to their objectives, technical methods, performances, and advantages and disadvantages. Comparative evaluations are provided to illustrate the influence of different sensor network settings on the target counting accuracy. The applicable environments of these algorithms are also discussed at the end of the chapter.

1 Introduction

Wireless sensor networks (WSNs) have been widely adopted to monitor various activities in many different types of environments due to the low manufacturing and maintenance cost they possess and the powerful functionality they provide. Some of the most important tasks include detecting the presence of a target of interest

D. Wu (✉) · B. Zhang · H. Li · X. Cheng
Computer Science, The George Washington University, Washington, DC 20052, USA
e-mail: andrewwu@gwu.edu

B. Zhang
e-mail: bowuzh@gwu.edu

H. Li
e-mail: hongjuan@gwu.edu

X. Cheng
e-mail: cheng@gwu.edu

H. M. Ammari (ed.), *The Art of Wireless Sensor Networks*,
Signals and Communication Technology, DOI: 10.1007/978-3-642-40066-7_7,

[1–5], counting the number of targets within a monitored area [6, 7], and estimating the locations of the targets [8–13], just to name a few. In extreme circumstances, such tasks can not be simply accomplished by human beings because the working environment may be unreachable or dangerous, or the labor cost is unacceptable.

One of the most fundamental problems in wireless sensor networks is to estimate the total count of the targets in a particular monitored region. In fact, counting is usually the first step in many other applications such as target tracking and positioning. For example, one may be interested in the group size, location, and mobility patterns of a group of sea birds in a certain seashore area for a given time period to infer other useful information. In this case, a sensor network is usually deployed to periodically gather relevant data. Note that the word "target" could refer to not only materialized objects but also some activities such as the "power-on" status of a device or a signal in transmission. Although it is usually difficult to obtain the exact target count number, existing methods have made a great effort to improve the counting accuracy. In this chapter, we will introduce and summarize the most recent and representative works tackling the problem of target counting in wireless sensor networks.

The main challenge of target counting in sensor networks lies in two-fold. On one hand, in order to fully cover an area of interest, sensors are usually densely deployed, which unavoidably results in overlaps in the sensing region. In other words, The presence of a target may be detected by more than one sensor. On the other hand, all the targets within the sensing range of a sensor can impact on the sensor's measurement—for example, a certain type of sensor may sample the strength of the accumulated target signals at the position the sensor resides; consequently, it is usually difficult to infer the exact target count from the sensor measurements in a straightforward way.

In this chapter, we provide a comprehensive survey of the most recent target counting techniques. To clearly state the developing trail of these approaches, we classify existing works into four major categories based on the sensing capabilities and estimating methods. *Binary counting* assumes that each sensor is able to report whether there exists at least one target in its sensing region, i.e., report a value "1" if any target is detected and '0' otherwise. *Numeric counting* takes it one step further: it assumes that each sensor reading captures the total number of targets covered by the sensing range of the sensor. *Energy counting* and *compressive counting*, on the other hand, employ the direct measurement of the target energy at each sensor to estimate the target count. These two categories differ in their estimation methods: compressive counting is fundamentally different from energy counting approaches as it exploits the *Compressive Sampling theory* to estimate the count of the targets while energy counting approaches make use of traditional machine learning and statistical techniques. The objective, mechanism, performance, and pros and cons of representative and the most important works are to be examined in detail. A comparative analysis will also be provided to demonstrate the applicable environments of these algorithms.

The rest of the chapter is organized as follows. In Sect. 2, we present an overview on the problem of target counting in sensor networks and provide a classification of the most recent target counting techniques. Following that we break down the

high-level classification and describe the corresponding techniques in detail. More specifically, we outline the various technical approaches based on binary counting, numeric counting, energy counting, and compressive counting in Sects. 3, 4, 5, and 6, respectively. Finally, open issues are discussed in Sects. 7 and conclusions are drawn in 8.

2 Overview

Although a number of target counting algorithms have been proposed for wireless sensor networks, each of them has its own cons and pros, and in particular, is suitable only for some operating environments. Before deploying any target counting algorithm for real world applications, we need to consider the following factors. First, due to the characteristics of different targets, different network models should be considered. For example, some targets may have visible contours, which can be detected by numeric photoelectric sensors [14]. For such targets, it is preferable to deploy a numeric counting sensor network. For the targets that can not be detected visually, such as the activities of some radioactive devices, energy sensors are usually preferred. Second, the target distribution and the sensor deployment pattern may vary. A target counting algorithm may yield precise counting results only in some specific scenarios. Last but not the least, the communication cost may be different. The power storage in a sensor is limited. As a result, it is preferred to deploy distributed algorithms that can save communication cost compared with centralized ones, in order to achieve a longer lifespan of the wireless sensor network. However, centralized algorithms tend to yield more precise counting results especially in situations where the targets are distributed densely and non-uniformly. Therefore sometimes we need to strike a balance between power consumption and counting accuracy.

In order to provide a clear view over the developing trail of different target counting methods, we classify the most recent works into the following four categories based on the sensing capabilities/network models and the theoretical foundations: *binary counting*, *numeric counting*, *energy counting*, and *compressive counting*. The binary counting model relies on binary sensors [15], which can detect the existence of one or more targets within its sensing range, and output a value " 1" (denoting the presence of one or more targets) or "0" (denoting the absence of any target). Since the output is only one bit, it has a low communication overhead and achieves a good robustness against noise. In this model, a lower bound of the target count can be obtained through geometrical analysis. The numeric counting model deploys numeric sensors that can figure out the number of targets within its sensing range by photoelectric technology [14]. Such local counts are then aggregated in a centralized server at which a global estimation of the target count can be generated. Energy counting relies on energy sensors [16], which can measure the signal strength of the target energies. Target count is estimated via analysis of the target energy property and the sensors' reading map. Compressive counting also relies on energy sensors. However, it counts the targets through sparse signal recovery based on the compressive sampling theory. In the following sections, we will provide a more detailed analysis on these methods.

3 Target Counting Based on Binary Sensors

3.1 Problem Formation

Binary sensors are widely used in target counting because of the simplicity and reliability they possess. A binary sensor outputs a reading of "1" when there exist targets within its sensing range, and '0' otherwise. Compared to other target counting techniques, binary counting is simple, practical, and robust. Because each sensor only needs 1 bit for the output, binary counting has the lowest communication cost for each communication session. Consequently, the power consumption is less compared with other methods. In addition, since the output of a binary sensor is either "1" or "0", it is easy for sensors to smooth out noisy readings. Because of the low communication cost and the robustness against noise, active research has been conducted to explore complex applications, such as target tracking, based on binary sensor networks [17–20].

The current binary target counting methods mainly focus on the following two problems:

1. How does a binary sensor detect the presence of targets by energy measurement?
2. Given the readings of the binary sensors in the monitored area, how to combine them and estimate the count of the targets in the whole network?

We will discuss each of the above problems and summarize the corresponding major technical approaches in the following subsections.

3.2 Target Detection

The authors in [21] propose to use hypothesis testing for the problem of detecting the presence of targets within a binary sensor's region of interest (ROI). In their work, the target detection process is modeled as an independent binary decision between:

$$H_1 : r_i = e_i + n_i \tag{1}$$

and

$$H_0 : r_i = n_i \tag{2}$$

where r_i is the reading of the sensor s_i, e_i is the signal strength at s_i's location, and n_i denotes the noise level with a zero mean Gaussian distribution. The hitting rate p_{h_i} and the false alarm rate p_{f_i} of s_i can be defined as:

$$p_{h_i} = \int_{\tau}^{\infty} \frac{1}{2\pi} e^{\frac{-(x-e_i)^2}{2}} dx \tag{3}$$

$$p_{f_i} = \int_{\tau}^{\infty} \frac{1}{2\pi} e^{\frac{-x^2}{2}} dx \tag{4}$$

where τ is the parameter used for adjusting the hitting rate and false alarm rate, and is assumed to be the same for all the sensors.

To further enhance the detection accuracy, this work [21] also proposes an approach to integrate the local decisions. Suppose that in a local group consisting of n sensors, each sensor s_i makes an independent local decision $D_i (i = 1, 2 \ldots, n)$, a more precise decision can be made by comparing $\sum_{i=1}^{n} D_i$ with a threshold T. Assume that the detection rate and the false alarm rate of sensor s_i are p_{h_i} and p_{f_i}, respectively. To achieve a better detection rate and a lower false alarm rate than the individual decision, an interval of T is derived from the Chebyshev's inequality, which is presented as follows:

$$\left[\sum_{i=1}^{N} p_{f_i} + \sqrt{\sum_{i=1}^{N} (1 - p_{f_i})}, \sum_{i=1}^{N} p_{h_i} - \sqrt{\sum_{i=1}^{N} p_{h_i}} \right] \tag{5}$$

The proposed method provides a theoretically sound solution to the following fundamental problem in target counting: how to detect the presence of the targets. However, no information about e_i is available in real applications, since we do not even know whether there is a target within the monitored area. The value of e_i is usually replaced with the smallest possible signal amplitude that can be detected by the sensor s_i if a target is actually within s_i's sensing region. As a result, the hitting rate and the false alarm rate are only approximately determined.

3.3 Multiple Target Counting

Because the information obtained from binary sensors is limited, it is difficult to infer the exact target count or provide an unbiased estimator for the number of targets within the monitored area. Therefore obtaining the lower bound and/or the upper bound of the target count attracts a lot of attention. In [22], the authors provide a lower bound on the target count for binary counting based on geometric analysis. They also point out that the difficulty of binary counting lies in the interference on the sensor's readings caused by multiple targets other than the dimension of the space. Therefore, the paper considers the one dimensional target counting problem as an example. Assume that sensors are deployed along a line and each sensor's sensing region is represented by an interval C_i with a length $2R$. Then it is argued in [22] that no matter how close two targets are, they can always be distinguished if there are two sensors with non-overlapping sensing regions that can detect them. In the proposed geometric analysis, sensors are divided into two groups: on-sensors with output "1" and off-sensors with output "0." Geometrically, the on-sensors tell the regions where the targets might reside, and the off-sensors tell the regions where

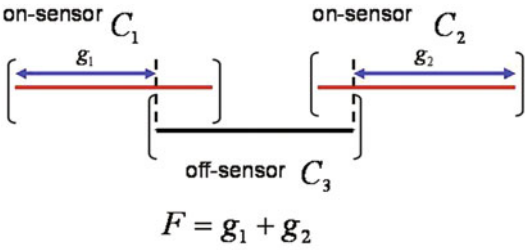


Fig. 1 An example of the feasible target space F

there is no target. The feasible target space F is defined as the union of the sensing regions of the on-sensors removing the overlapping areas with the sensing regions of the off-sensors:

$$F = \sum_{i \in \text{on-sensors}} C_i - \sum_{j \in \text{off-sensors}} C_j \tag{6}$$

Figure 1 shows an example of a feasible target space. Two sensors are said to be *positively independent* if they both report "1" when either their sensing regions are disjoint or they belong to different connected components of the feasible target space. Then the lower bound of the target count is consistent with the maximum number of the positively independent sensors. In other words, when all targets are well separated, the lower bound can be regarded as the target count.

The method presented above provides a lower bound for the number of targets in a given region. This lower bound can serve as a rough estimation for the target count when the targets are well separated. However, the bound may be too loose for dense targets. In addition, this lower bound does not increase with the increase of the number of targets after all sensors output "1." As a result, the proposed method is appropriate for sparse target counting only.

4 Target Counting Based on Numeric Sensors

4.1 Problem Formation

With the development of the wireless sensing technologies, sophisticated numeric sensors such as the photoelectric-based sensors are designed. These sensors can output the target counts within their sensing regions with a high accuracy.

The numeric counting model can be simply summarized as follows. Assume that there are N_s sensors deployed in the grid points or randomly and uniformly in a two-dimensional monitored area. The set of sensors are represented by $S = \{s_1, s_2, \ldots, s_{N_s}\}$. Let $x_i = (x_1^{(i)}, x_2^{(i)})$ denote the location of the sensor s_i, $i = 1, 2, \ldots, N_s$. The set of all sensors' locations are represented by

$L_S = \{x_1, x_2, \ldots, x_{N_s}\}$. The sensing region of s_i, denoted by $A(s_i)$, can have an arbitrary shape such as a circle, a square, an eclipse, or other irregular shapes due to the specialties in its design. But we consider a circular region centered at x_i with a radius h here for simplicity. The sensor s_i outputs a reading r_i, which is the count of the targets residing in s_i's sensing region $A(s_i)$. Let $R_S = \{r_1, r_2, \ldots, r_{N_s}\}$ denote the set of all such readings from the corresponding sensors. The monitored area $\mathcal{A}$ can be defined as the union of the sensing regions of all sensors.

In order to completely cover the area of interest, overlapping of different sensing regions is unavoidable. As a result, a target in the overlapping area may be detected by more than one sensor. This phenomenon results in *double counting*, in which a target may be counted at least twice. In order to obtain an accurate target count, the count of targets in the overlapping sensing regions should be analyzed carefully .

4.2 Geometry-Based Target Counting

In [23], the authors propose a method for the target counting problem based on geometric analysis. To reduce the interference of overlapping regions, the authors start with selecting a non-redundant subset of the sensors $S' \in S$. And the union of the sensing regions of S' equals R, while no sensors can be deleted from S' without losing any coverage of the monitored area. Figure 2 illustrates a simple network with three sensors, one of which is redundant as its sensing region is covered by the other two sensors.

We assume that $R' = (r_{1'}, r_{2'}, \ldots, r_{n'})$ denotes the reading set of the non-redundant sensor subsets; then the target count can be estimated by the following equation:

$$\hat{N}_t = \frac{\sum_{i=1}^{n'} r'_i}{\sqrt{m}} \tag{7}$$

where m represents the maximum degree of overlapping in S'. It is straightforward to conclude that the true target count should be less than the sum of the readings from R', $\sum_{i=1}^{n'} r'_i$, and more than $\frac{\sum_{i=1}^{n'} r'_i}{m}$, considering that the maximum degree of overlapping is m. As a result, the true value of the target count can always be bounded

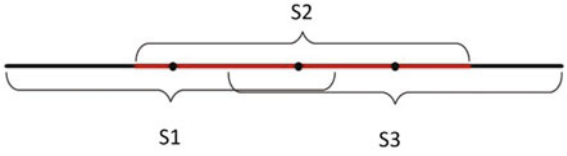


Fig. 2 An example network with a redundant sensor $s2$

by the range $[\frac{\hat{N}_t}{\sqrt{m}}, \sqrt{m}\hat{N}_t]$. Then target count estimator $\hat{N}_t$ can be regarded as the geometric mean of the two bounds.

In this work, the authors also extend their proposed estimation method to non-ideal sensing cases. If the count of targets around a sensor is c, and the sensor can report any value in the range of $[(1-\rho)c, (1+\rho)c]$, where ρ represents the noise level, then the true target count in the monitored area can be estimated by the following equation:

$$\hat{N}_t = \frac{\sum_{i=1}^{n'} r'_i}{\sqrt{m(1-\rho^2)}} \tag{8}$$

And the true target count can be bounded by the following range:

$$\left[\frac{\hat{N}_t}{\sqrt{m\frac{1+\rho}{1-\rho}}}, \sqrt{m\frac{1+\rho}{1-\rho}}\hat{N}_t\right]$$

This proposed method is simple and efficient. The computational complexity to build the non-redundant sensor subset is $O(N_s log(N_s))$ for 1D networks. For two-dimensional networks, the computational complexity is still polynomial. Besides, the authors also consider the non-ideal sensing model and extend the corresponding solution to the target counting problem in noisy environments. However, this method works well only for the cases in which both the sensors and the targets are deployed in a uniform pattern. Otherwise, the estimation may deviate significantly from the true target count. In addition, the proposed target count bounds are loose and can only provide limited information about the true target count.

4.3 A Probability Mass Function Approach

In [14], the authors propose a target counting approach based on probability theory. They start with deriving a probability mass function (pmf) of the target distribution. Assuming that the targets are uniformly distributed in the monitored area, then the probability that there are k targets in $A(s_i)$ of size S can be computed as follows:

$$P(r_i = k) = \frac{e^{-\gamma S}(\gamma S)^k}{k!} \tag{9}$$

In the above equation, γ represents the density of the targets, which can be regarded as a known or unknown factor. If it is unknown, it can be approximately estimated from the sensor's reading map.

In the next step, the authors partition the monitored area into smaller subareas following a simple procedure: the boundaries of the sensing regions of all sensors provide a natural cut to partition the area. As illustrated in Fig. 3, the area is partitioned

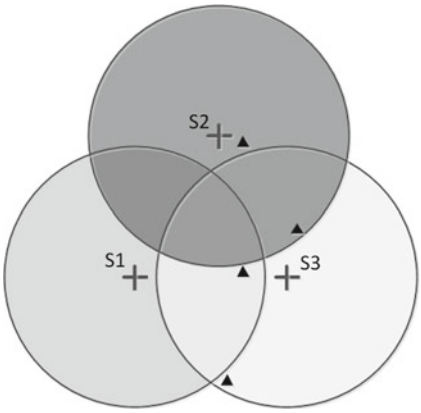


Fig. 3 Partition the monitored area

into seven subareas by the boundaries of three sensors, with each subarea being covered by one or more sensors.

Note that the subareas are not overlapped. Denote by $M = \{m_1, m_2, \ldots, m_o\}$ the set of subareas, and by r_{m_i} the target count in m_i, supposing that there are o different subareas in total. If r_{m_i} is regarded as a random variable, it is reasonable to assume that each r_{m_i} is independent. As a result, the following conclusion can be reached

$$P(r_{m_i} = k_1 \cap r_{m_j} = k_2) = p(r_{m_i} = k_1) \times p(r_{m_j} = k_2) \tag{10}$$

On the other hand, based on the definition of conditional probability, the following equation holds:

$$P(N_t = k|R_S) = \frac{\sum_{R_m \in A \cap B} P(r_{m_1} = k_1, r_{m_2} = k_2, \ldots r_{m_o} = k_o)}{\sum_{R_m \in A} P(r_{m_1} = k_1, r_{m_2} = k_2, \ldots r_{m_o} = k_o)} \tag{11}$$

In the above equation, A and B represent the following constraints:

$$A : \begin{cases} \sum_{m_i \in M_{s_1}} r_{m_i} = r_1 \\ \sum_{m_i \in M_{s_2}} r_{m_i} = r_2 \\ \ldots \\ \sum_{m_i \in M_{s_{N_S}}} r_{m_i} = r_{N_S} \\ r_{m_i} >= 0; \end{cases} \tag{12}$$

$$B : \sum_{i=1}^{o} r_{m_o} = k \tag{13}$$

Given the sensors' readings, the target count in the monitored area can be estimated by its conditional expectation, which can be represented by the following equation:

$$E(N_t|R_S) = \sum_{t=\min}^{\max} t \times p(N_t = t|R_S) \tag{14}$$

where min and max represent the smallest possible and the largest possible target counts, respectively.

This estimator sounds reasonable and it can provide an accurate estimation when targets are uniformly distributed. However, the computational complexity is not polynomial since the computation of the above equations needs to enumerate all possible values of the target count as well as all possible target count values in each subarea.

In order to make the algorithm practical, the authors propose the following heuristics in [14]. First, if the sensing region of a sensor contains no target, the sensor as well as its sensing region will not be considered. Second, because the computational complexity is not polynomial, the computation cost can be reduced significantly by partitioning the monitored region into smaller ones and estimating the target count in each sub-region independently. Thus, the authors introduce a network partition algorithm. A sensor network can be modeled as a topology graph $G(V, E)$, where V represents the set of N_s sensors as well as their sensing regions, and an edge e_{ij} exists between sensors s_i and s_j if and only if their sensing regions overlap. The objective of the algorithm is to find a partition that can result in two balanced sub-partitions with the interference between them minimized. An objective function, which can reflect the interference between two sub-partitions, is defined as follows:

$$f_{obj}(G_1, G_2) = \sum_{s_i \in G_1, s_j \in G_2} size(A(s_i) \cap A(s_j)) \times (r(s_i) + r(s_j)) \tag{15}$$

The graph partition process intends to seek a balanced partition G_1 and G_2 with a minimum $f_{obj}(G_1, G_2)$ value, and is described as follows:

1. Randomly generate a partition G1 and G2, each of which is composed of half of the sensors and their sensing regions.
2. Select a node in G1 and a node in G2 such that switching these two nodes results in the most significant decrease in the objective function value. Switch these two nodes and lock them. A locked node will not be considered for switching again.
3. Repeat step 2 until no further decrease in the objective function value can be achieved by switching two unlocked nodes.

This heuristic partition algorithm can be completed in polynomial time and can result in a sub-optimal partition. Figure 4 shows an example to illustrate the partitioning procedure. As shown in Fig. 4a, a random partition is first generated and the corresponding objective function value is 316. After exchanging s_2 in G_2 with s_6 in G_1, the new partition achieves an objective function value of 65 (Fig. 4b).

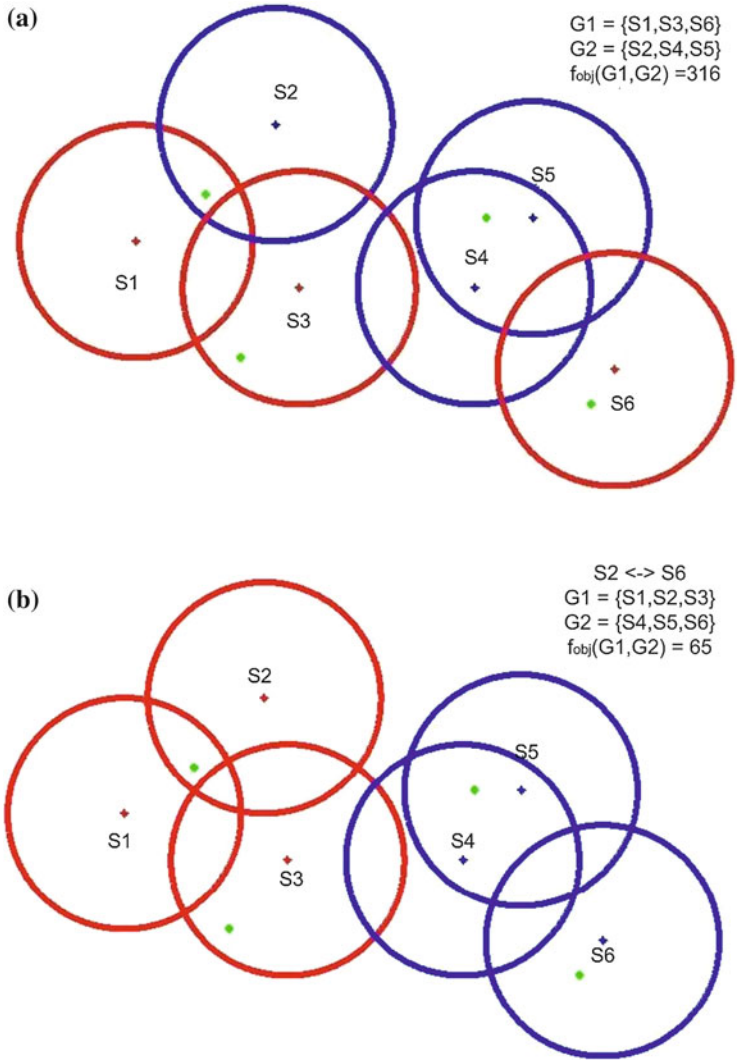


Fig. 4 Partition a graph into subgraphs. **a** A random partition. **b** Exchange two nodes

By recursively applying the partition algorithm several times, the sensor graph can be partitioned into a few sub-graphs, with each having at most a pre-defined number of sensor nodes. The sum of the estimated target counts from all sub-graphs is considered as a rough estimation of the total target count. This approach has one drawback: the targets in the overlapping areas of two sub partitions are still duplicately counted and may result in over-estimation. In order to solve this problem, two compensation schemes, namely a *minus compensation scheme* and a *plus compensation scheme*, are provided to tune the estimation.

The proposed approach summarized above is the first attempt to solve the double counting problem based on probability theory. It works for the cases when the targets are uniformly distributed in the whole area or piecewise uniformly distributed (uniformly distributed within each subarea). Precision can be guaranteed for both cases. However, the time complexity is high and the computation time is unacceptable when the network topology is complex and the count of targets is large. Although the network partition algorithm can reduce the computation time, the time complexity is still not polynomial. As a result, this approach can not be applied to large scale sensor networks.

4.4 An Applied Statistical Approach

In [24], the authors tackle the double counting problem from a different perspective. In sensor networks, it is easy to find a subset of sensors that have no overlapping sensing region. If the target distribution is known, the total number of targets can be estimated using statistical methods from readings of non-overlapping sensors. This procedure can be repeated multiple times, i.e., multiple subsets of sensors, with each containing sensors with non-overlapping sensing regions, can be considered independently, yielding a sequence of estimated target counts. The final target count can then be computed from these estimated results based on the maximum likelihood estimation.

The authors in [24] start with estimating the distribution of the targets. Assume that the target position distribution is $f(X|\theta)$, which can be partially known or unknown, regular or irregular. The distribution in two-dimensional space can be estimated based on the sensors' readings. Given the assumption that the targets are identically and independently distributed, based on the definition of conditional expectation in probability theory and the property of continuous functions in calculus, it is proved that the following approximation holds:

$$E(Y \mid x) \approx N_t \times f(x|\theta) \tag{16}$$

where $Y = \frac{R}{\pi h^2}$, with R denoting the count of the targets in the circular region centered at x with a radius h. This approximation shows that the expected count of the targets per unit area at x is approximately proportional to the value of $f(x|\theta)$ at x. Therefore, an approximate target distribution function can be obtained using the regression techniques.

Depending on whether or not certain prior knowledge is available regarding the target distribution, different regression techniques can be employed. For the case that the type of target distribution is known while the parameters of the distribution are unknown, a parametric regression technique can be taken to estimate the unknown parameters. In [24] the following parametric model is used:

$$R_S = c \times f(X|\theta) \tag{17}$$

An estimate $\hat{\theta}$ of θ can be obtained by minimizing the sum of residue squares, i.e., by solving

$$\min_{\theta} \sum_{i=1}^{N_s} (r_{s_i} - c \times f(X_i|\theta))^2, \tag{18}$$

The Levenberg-Marquardt algorithm (LMA) [25] is usually employed to find a sub-optimal solution $\hat{\theta}$ to this minimization problem.

For the case with no prior knowledge about the target distribution, a non-parametric kernel regression model is applied to estimate the target distribution. The non-parametric model first estimates the expected number of targets per unit area $D(x_1, x_2)$ in the monitored area using the Nadaraya-Watson estimator [26], which can be represented by the following equation:

$$D(x_1, x_2) = \frac{\sum_{i=1}^{N_s} K\left(\frac{x_1 - x_1^{(i)}}{h}\right) K\left(\frac{x_2 - x_2^{(i)}}{h}\right) r_{s_i}}{\sum_{i=1}^{N_s} K\left(\frac{x_1 - x_1^{(i)}}{h}\right) K\left(\frac{x_2 - x_2^{(i)}}{h}\right)}$$

where $K()$ is a Gaussian kernel function. With the knowledge of the expected target count per unit area at each point, by scaling $D(x_1, x_2)$ and making the integration of $D(x_1, x_2)$ over the monitored area to 1, the approximate target distribution can be obtained.

Given the the approximate target distribution, a maximum likelihood estimator for target counting can be applied to subsets of sensors with non-overlapping sensing regions. Obviously, if the sensing regions of a set of sensors do not overlap, the sum of their readings precisely reflects the total count of the targets within their coverage areas. A non-overlapping sensor subset selection process is described as follows. Let S_i be the subset selected by the ith run. Initially, $S_i = \emptyset$.

1. Randomly select a sensor s_j, and add s_j to S_i.
2. Mark sensors whose sensing regions overlap with that of s_j. These marked sensors will not be considered in the same sensor subset selection process.
3. Repeat steps 1 and 2 until a predefined number of sensors are selected or no more sensors can be selected.

Figure 5 illustrates an example of the selection of a subset of sensors with non-overlapping sensing regions. The selection process takes linear time.

Suppose that the non-overlapping sensor subset sampling process is executed for n times, and the union of the sensing regions of the ith subset is represented by $A(S_i)$. Assume that targets are i.i.d, it is straightforward to reach the following conclusion: the number of targets that falls into $A(S_i)$ can be regarded as a binomial random variable $B(N_t, P_i)$, where N_t is the unknown target count in the monitored area, and P_i can be estimated by the following equation:

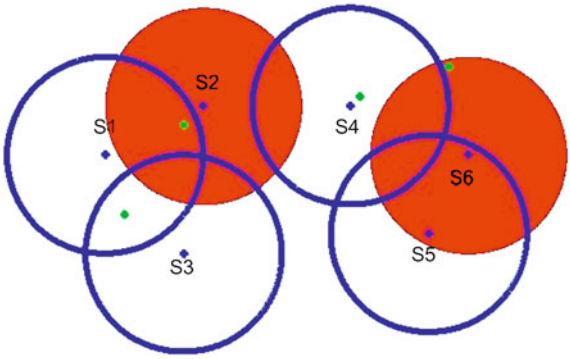


Fig. 5 An example of non-overlapping sensor subset

$$\begin{aligned}\hat{P}_i &= \int\int_{\cup A(s_j), s_j \in S_i} \hat{f}(x|\theta)dx_1dx_2 \\ &= \sum_{s_j \in S_i} \int\int_{A(s_j)} \hat{f}(x|\theta)dx_1dx_2\end{aligned} \tag{19}$$

If n different sensor subsets are selected, and the count of the targets that fall into the sensing regions of these sensors are recorded as $N_{t1}, N_{t2}, \ldots, N_{tn}$, the target count N_t can be estimated by the maximum likelihood estimation (MLE) based on the binomial assumption. The likelihood function is described as follows:

$$l(N_t) = \prod_{i=1}^{n} \binom{N_t}{u_i} \hat{P}_i^{u_i} (1 - \hat{P}_i)^{N_t - u_i} \tag{20}$$

The value of $\hat{N}_t$ that maximizes the likelihood function is the estimated target count. Traditional method solving the MLE [27] involves computations of differentiation, which is extremely time-consuming due to the factorials in this context. However, it is proved in [24] that (20) has only one local maxima value for this context. In addition, when $N_t < \hat{N}_{tmle}$, $l(N_t)$ is an increasing function; when $N_t > \hat{N}_{tmle}$, $l(N_t)$ is a decreasing function. As a result, the authors propose a heuristic to try from the smallest possible target count to the largest possible target count. The first N_t that satisfies $(l(N_t) - l(N_t + 1)) > 0$ is taken as $\hat{N}_{tmle}$. With this heuristic, the problem can be solved efficiently.

Compared with the algorithm reviewed in Sect. 4.3, the statistical counting approach can be applied to more patterns of targets. In addition, the computation takes less time.

5 Target Counting Based on Energy Sensors

5.1 Problem Formation

Targets emit energy signals such as acoustic, light, or heat. Energy sensors, which can sense a specific type of signals from surrounding targets, have the potential to be used to estimate the count of targets within a monitored area.

The problem of target counting based on energy sensors is described as follows. Let $S = \{s_1, s_2, \ldots, s_{N_s}\}$ be the set of N_s sensors deployed in a two-dimensional monitored area R, and $T_g = \{t_1, t_2, \ldots, t_{N_t}\}$ be the set of N_t targets who have an arbitrary distribution. When there is no ambiguity from context, let $s_i = (x_{s_i}, y_{s_i})$ also represent the location of the sensor s_i and $t_i = (x_{t_i}, y_{t_i})$ also represent the location of the target t_i. It is widely accepted that the emission of the target signal follows a decay model that can be captured by a function $f_{\mathcal{E}}(d)$, which is a decreasing function of d, the Euclidian distance to the target. Although the decay model may vary in different papers, most researchers use a model of $f_{\mathcal{E}}(d) = \frac{E_t}{(1+d)^{\alpha}}$, with α being the decay factor in the range of $[2.0, 5.0]$ [28, 29]. The signal amplitude $\mathcal{E}(p, T_g)$ at any point p in R is the superposition of the signal amplitudes of all targets at the position p, which can be written as:

$$\mathcal{E}(p, T_g) = \sum_{j=1}^{N_t} f_{\mathcal{E}}(d(p, t_j)). \tag{21}$$

The reading of sensor s_i is $\mathcal{E}(s_i, T_g)$ in a noise free environment. The task of the energy-based target counting is to estimate the count of targets based on sensor energy readings and the location information.

Note that although in real world, the base energy levels are different for different targets, in this category, only identical targets are considered by the existing research.

5.2 Distributed Aggregate Management

The authors in [16] present a simple target counting protocol for wireless sensor networks. Assume that targets are identical and sparsely deployed. If the reading of a sensor $\mathcal{E}(s_i, T_g)$ is significantly higher than those of its neighbors, there should be a target near s_i , and the target should be closer to s_i than to s_i's neighbors. Therefore, the count of the targets can be estimated by the count of the sensors with local maximum readings. This idea is explored by the distributed aggregate management (DAM) protocol, which is stated as follows.

In DAM protocol, each sensor records the following data fields: the largest sensor reading (MaxPr) it has ever seen in a counting procedure, the ID of the sensor (MaxID) with the largest reading, the ID of its one hop parent (transID), and the reading of

its parent (transPr). Every data packet includes all the above information about its sender as well as the reading of the sender. A sensor that has a higher reading than all its neighbors is called a leader. A parent of a sensor is its one hop neighbor that relays a packet from the leader to the sensor. The process of identifying a target coincides with the leader election process. The leader election process is performed periodically and can be described as follows:

1. At the beginning of each leader election period, sensors with larger readings than a threshold δ_1 will all join the leader election process. The MaxPr and transPr of the sensor are initialized to be its own sensor reading. The MaxID and transID of a sensor are initialized as its own ID. Each sensor will broadcast its state information to all its one hop neighbors.
2. Once receiving a data packet, a sensor checks if the MaxPr in the packet is larger than its recorded MaxPr and if its own reading is smaller than the reading of the packet sender. If both conditions are satisfied, the sensor first updates its state based on the received information by resetting MaxPr, MaxID, transPr, and transID, and then broadcasts MaxPr, MaxID, and its own ID and reading to all its neighbors. Otherwise, the received packet is dropped.

The above process needs to be executed during each counting period. A counting period ends when no sensor needs to update its state and no more broadcast occurs. Finally, the sensors whose MaxPr and transPr are equal to their own readings claim themselves as leaders and report to the server. Figure 6 illustrates the leader election process in DAM. This example contains 9 targets and 100 sensors represented by solid circles and "+" signs, respectively, in an area of 100 m × 100 m. The solid squares

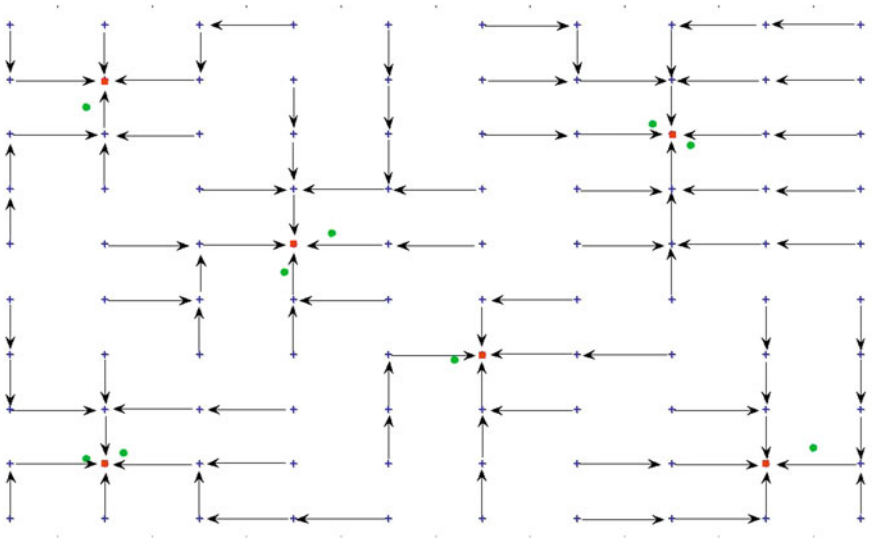


Fig. 6 Leader election in DAM

represent the sensors that are elected to be leaders by the algorithm. When the leader election procedure converges, six leaders are elected, representing six clusters. Each cluster is a tree rooted at its elected leader.

This protocol is energy efficient and has a fast convergence. However, an accurate estimation can be achieved only when the targets are sparsely distributed or well separated. In addition, the authors argue that for grid sensor deployment, two targets can be discerned clearly by the protocol if $d > 2.5r$, where d represents the distance between the two targets and r represents the distance between two neighbor sensors. While the counting precision of DAM can only be guaranteed when the targets are well separated, it forms the foundation of a series of other methods.

5.3 Energy-Based Activity Monitoring

When targets are densely deployed, it is not always possible to separate the signals from two targets that are close to each other using DAM. To address this issue, the authors in [30] propose a protocol termed EBAM, which stands for energy-based activity monitoring. EBAM first runs the DAM protocol. Sensors that share the same leader form a group. The number of targets that reside in the coverage area of each group is estimated independently using the following proposed method.

It is straightforward to observe that if the signal landscape $\mathcal{E}(p, T_g)$ over the monitored area is known, the energy volume of all targets can be computed by the following equation:

$$EV = \int_{p \in R} \mathcal{E}(p, T_g) d(p) \tag{22}$$

Since the unit target energy volume (UEV) can be obtained by the integral of the signal decay function, the target count could be estimated by $\frac{EV}{UEV}$. However, due to the fact that only the signal amplitude at a sensor's location is available, a signal landscape recovery method based on the sensor readings is needed. EBAM employs an estimated signal landscape based on the Voronoi graph model [31].

The Voronol Cell of a node s_i, denoted by $V_c(s_i)$, consists of all the points that have a shorter distance to s_i than to any other nodes, i.e. a point $p \in V_c(s_i)$ if and only if $d(p, s_i) \leq d(p, s_j)$ for any $s_j, s_j \neq s_i$. Figure 7 shows the Voronoi graph of 25 sensors in a monitored area. Given $\mathcal{E}(s_i, T_g)$, the estimated energy landscape over $V_c(s_i)$, denoted by $\hat{\mathcal{E}}_{vc}(s_i, T_g)$, can be built by setting the signal amplitude of each point in $V_c(s_i)$ to $\mathcal{E}(s_i, T_g)$, i.e., for $\forall p \in V_c(s_i)$, $\hat{\mathcal{E}}(p, T_g) = \mathcal{E}(s_i, T_g)$. Then the estimated energy landscape over the whole monitored area, denoted by $\hat{\mathcal{E}}_R(S, T_g)$, is expressed as:

$$\hat{\mathcal{E}}_R(S, T_g) = \cup \hat{\mathcal{E}}_{vc}(s_i, T_g) \tag{23}$$

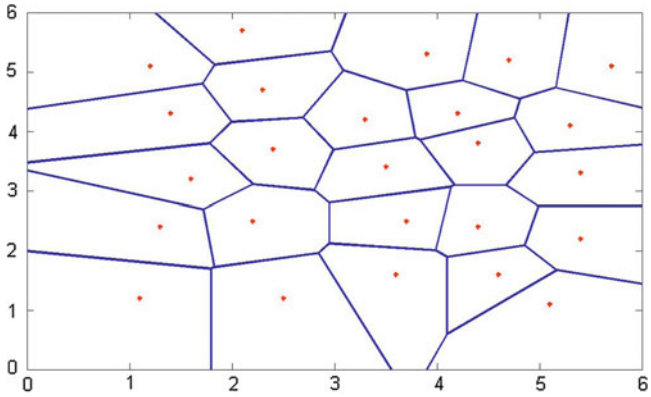


Fig. 7 A Voronoi diagram for 25 sensors in a monitored area

Using the integral of the estimated energy landscape over the monitored area, the estimated energy volume ($\hat{EV}$) can be obtained. Consequently, the estimated target count in each group can be derived as $\frac{\hat{EV}}{UEV}$.

Due to the attenuation effect of target signals, a sensor's reading may vary dramatically even if the position of a target changes a little. To tackle this issue, a ceiling value is introduced in EBAM to limit the maximum contribution of a single sensor to the local sensor group.

Compared with DAM, EBAM can be utilized for dense target deployment. However, there are some factors that EBAM fails to consider. First, the authors do not consider the energy leaking effect. If a target is located at the boundary of the monitored area, part of the energy may not be included in the energy volume estimation in EBAM. Second, the introduced ceiling factor helps to provide a steady target count estimation only if the targets are very close to a certain sensor. No proper compensation is provided for the case when a target is far away from all sensors. As a result, the estimation may be biased.

5.4 Energy-Based Target Enumeration

In [29], the authors propose an energy-based target enumeration protocol (EBTN) based on DAM and EBAM. EBTN aims to improve the communication efficiency and counting precision. In the leader election phase, EBTN follows the DAM process except that once a sensor realizes that its reading is the largest among all its neighbors, it declares itself as a leader and broadcasts its position information to its neighbors immediately. Other sensors, with a reading larger than a threshold δ_2, will join the nearest leader automatically without incurring any data transmission. This procedure

achieves leader election and group formation with less time and less communication overhead compared to DAM.

In addition, EBTN improves the counting precision in the following two points: first, it builds a more accurate estimated signal landscape; Second, it compensates the energy leaking. Compared with the true energy landscape, the estimated signal landscape based on the Voronoi graph model is not smooth and less accurate, which may result in a large error in energy volume estimation. To solve this problem, EBTN employs a polynomial regression technique to build a more accurate estimated signal landscape. In EBTN, after the leader election and sensor group formation phase, each group leader collects the node location and signal amplitude information from its members. Let (x, y) represent a position in the two-dimensional monitored area. The following parametric polynomial model is employed to estimate the signal landscape over the monitored area covered by a group of sensors:

$$\mathcal{E}(x,y) = \beta_0 + \beta_1 y + \beta_2 y^2 + \beta_3 x + \beta_4 xy + \beta_5 xy^2 + \beta_6 x^2 + \beta_7 x^2 y + \beta_8 x^2 y^2 \quad (24)$$

where $\mathcal{E}(x, y)$ is the interpolated signal strength of any given point in the area of interest. Using the least residue square estimation, the parameter vector β in the above equation can be calculated by $\beta = (X^T X)^{-1} X^T Z$, with

$$X = \begin{pmatrix} 1 & y_{s_1} & y_{s_1}^2 & x_{s_1} & x_{s_1} y_{s_1} & x_{s_1} y_{s_1}^2 & x_{s_1}^2 & x_{s_1}^2 y_{s_1} & x_{s_1}^2 y_{s_1}^2 \\ 1 & y_{s_2} & y_{s_2}^2 & x_{s_2} & x_{s_2} y_{s_2} & x_{s_2} y_{s_2}^2 & x_{s_2}^2 & x_{s_2}^2 y_{s_2} & x_{s_2}^2 y_{s_2}^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & y_{s_n} & y_{s_n}^2 & x_{s_n} & x_{s_n} y_{s_n} & x_{s_n} y_{s_n}^2 & x_{s_n}^2 & x_{s_n}^2 y_{s_n} & x_{s_n}^2 y_{s_n}^2 \end{pmatrix} \quad (25)$$

where (x_{s_i}, y_{s_i}) represents the location of the ith sensor in the group, and Z denotes the signal amplitude vector with the ith element in Z corresponding to the reading of the ith sensor in the group. Based on the estimated signal landscape, the estimated energy volume can be obtained. Note that as pointed out by the authors, the energy volume is usually underestimated due to energy leaking. A constant factor of 0.7 is then multiplied to the unit target energy volume in their work, which yields a good accuracy according to the simulation study in [29].

Compared with the previously introduced DAM and EBAM, EBNT has the following advantages. First, the proposed leader election process converges faster, and incurs less communication overhead. Second, the estimated signal landscape based on the polynomial regression model can better recover the true energy landscape, which leads to a more accurate energy volume estimation. Third, the authors consider the energy leaking problem and provide a constant factor that can help to make a more precise target count estimation.

However, for different sensor deployment patterns and target distributions, the constant factor should be different. It may be difficult to find the right value for this compensation factor.

5.5 A Monte Carlo Method for Target Counting

From the previous section, we understand that the energy volume from the targets may be underestimated due to energy leaking. Thus an adaptive method is required to compensate the energy leaking for different sensor deployment patterns and target distributions. To study this problem, the authors in [28] propose an approach to employing Monte Carlo simulations for Target Counting (MCTC).

It is assumed in [28] that targets follow a uniform distribution or a distribution that can be approximately modeled as piecewise-uniform. The estimated energy landscape based on the Voronoi graph model [31] is employed. And the authors prove that for uniform or piecewise-uniform targets, when the number of targets becomes large, the distribution of the targets approaches to the estimated energy landscape closely after scaling. As a result, the scaled estimated energy landscape can be regarded as the approximate target distribution.

Based on this observation, the authors conduct Monte Carlo simulations [32] in a virtual monitored area. The virtual area R' is exactly the same as the real monitored area R. N_s virtual sensors, denoted by $S' = \{s'_1, s'_2, \ldots, s'_{N_s}\}$, are assumed to be deployed in R', with the position of s'_i being the same as that of s_i in R. N'_t virtual targets, denoted by $T'_g = \{t'_1, t'_2, \ldots, t'_{N'_t}\}$, are added to R' based on the estimated target distribution. The virtual targets have exactly the same target energy properties as the real targets. An Acceptance-Rejection sampling technique [33] is applied to control the virtual target deployment to make sure that the shape of the virtual energy landscape is as similar as possible to that of the real target energy landscape when the algorithm terminates. When the number of virtual targets N'_t becomes large, the distribution of the virtual targets approaches to that of the real targets. As a result, the energy leaking rate for the real targets and the virtual targets should be similar.

Given the virtual targets' locations and their energy decay model, the reading of each virtual sensor can be easily calculated. Specifically, the energy volume of $\hat{\mathcal{E}}_R(S, T_g)$, denoted by $\hat{V}_{\mathcal{ER}}(S, T_g)$, could be expressed as:

$$\hat{V}_{\mathcal{ER}}(S, T_g) = \sum_{i=1}^{m} \hat{\mathcal{E}}(s_i, T_g) \times Area(V_c(s_i)) \tag{26}$$

Through the same method, the estimated virtual target energy volume can also be easily obtained. Thus, the number of targets N_t can be estimated by the following equation:

$$N_t = \frac{V_{\mathcal{ER}}(S, T_g) \times N'_t}{V_{\mathcal{E}R}(S', T'_g)} \tag{27}$$

To achieve a more accurate estimation, the authors consider sensor readings in multiple epoches. They assume that each sensor reports its reading to the central server at every epoch and a fixed number of targets keep moving in the monitored area. Let T_{max} be the maximum number of epochs to be considered.

Then the estimated energy landscape at the T_{max} epoches is recorded as: $\hat{\mathcal{E}}_R^1(S, T_g)$, $\hat{\mathcal{E}}_R^2(S, T_g)$, $\ldots$, $\hat{\mathcal{E}}_R^{T_{max}}(S, T_g)$. Correspondingly, the energy volume for each estimated energy landscape, denoted by $\hat{V}_{ER}^1(S, T_g)$, $\hat{V}_{ER}^2(S, T_g)$, $\ldots$, $\hat{V}_{ER}^{T_{max}}(S, T_g)$, can be obtained. Similarly, in each epoch i, a virtual energy landscape $\hat{\mathcal{E}_R}^i(S', T_g')$ can be built as described earlier, and the corresponding estimated energy volume is recorded as $\hat{V}_{\mathcal{E}R}^i(S', T_g')$. Finally, the count of the targets can be estimated through the following equation:

$$\hat{N}_t = \frac{\sum_{i=1}^{T_{max}} \hat{V}_{\mathcal{E}R}^i(S, T_g) \cdot N_t'}{\sum_{i=1}^{T_{max}} \hat{V}_{\mathcal{E}R}^i(S', T_g')}. \tag{28}$$

The authors prove that the proposed estimation is approximately unbiased for uniform targets and piecewise uniform targets.

Compared with the methods described previously, MCTC provides a more accurate solution to the problem of target counting, and the solution is proved to be approximately unbiased for uniform and piecewise uniform targets. However, this method requires each sensor to communicate with the central server, which may incur more energy consumption compared with the distributed methods based on local communications.

6 Target Counting Based on Compressive Sensing

6.1 Fundamentals of Compressive Sensing

Compressive Sampling (CS) [34] is a newly developed sampling paradigm in data acquisition that can reconstruct a sparse signal with a much lower sampling rate compared to the traditional Nyquist rate.

The basic idea of CS can be described as follows. Let X be a $N \times 1$ column vector in R^N. Given an $N \times N$ orthogonal basis $\Psi = [\Psi(1), \Psi(2), \ldots, \Psi(N)]$ with each $\Psi(i)$ being a column vector, X can be expressed by (29),

$$X = \Psi\Theta = \sum_{i=1}^{N} \Theta_i \Psi(i), \tag{29}$$

where Θ is the coefficient sequence of X in the transform domain Ψ. The signal X is K-sparse if it is a linear combination of K basis vectors. That is, only K of the Θ_i coefficients are nonzero and the other $(N - K)$ ones are zero. If $K \ll N$, instead of acquiring N samples for X, one can reconstruct X by taking only a small set of measurements according to the Compressive Sampling theory:

$$Y = \Phi X = \Phi \Psi \Theta = A\Theta, \tag{30}$$

where Y is a $M \times 1$ vector, $K < M \ll N$, Φ is a $M \times N$ measurement matrix, and A is a $M \times N$ matrix. For a $N \times 1$ sparse vector Θ, it has been proved that if A holds the Restricted Isometry Property (RIP) [34], Θ can be recovered with only $M \geq O(Klog(N/K))$ measurements through ℓ_1-minimization at an overwhelming probability.

The definition of RIP is given below: a matrix A obeys RIP with parameters (K, δ) for $\delta \in (0, 1)$ if

$$1 - \delta \leq \frac{\|AV\|_2^2}{\|V\|_2^2} \leq 1 + \delta, \tag{31}$$

holds for all K-sparse vector V. It has been proved that A follows the RIP when Ψ is a typical transform basis such as Fourier or Wavelet and Φ is random [34].

If the measurement vector y is corrupted with noise, the measurement becomes

$$Y = A\Theta + \mathcal{N}, \tag{32}$$

where $\mathcal{N}$ is an unknown error term (e.g. an additive white Gaussian noise (AWGN)). Then the ℓ_1-minimization with relaxed constraints for reconstruction is

$$\min|\Theta|_{\ell_1} \text{subject to } \|A\Theta - Y\|_{\ell_2} < \epsilon \tag{33}$$

where ϵ bounds the amount of noise in the data. It has been proved [35] that the reconstruction error of Θ based on the value computed from (33) is bounded by $c_0\epsilon_0 + c_1\epsilon$, where c_0 and c_1 are small constants and ϵ_0 is the reconstruction error when Y is noiseless.

Motivated by the advances in CS, recent works [36–39] apply CS to sparse target counting, leading to a considerably low communication overhead due to the low sampling rate (less number of sensor measurements). These works view target locations as a sparse signal Θ and try to reconstruct Θ by CS with a small set of sensor measurements. Target counting methods based on the CS technique mainly consider the situation that targets are sparse compared to the total number of grids utilized to represent the locations of the targets. Target counting methods based on other models can also be applied to the same situation. However, CS is a fundamentally different methodology. In addition, CS can be applied to more complex situation when the energy model of different targets are not identical. Besides that, targets' location information can also be obtained from these methods.

One big challenge in CS based target counting is to justify the applicability of CS to the target counting problem. That is, whether A obeys the RIP. In addition, traditional CS recovery algorithms possess a high time complexity, while the sparse signal in target counting often has special properties that can be exploited to simplify

the recovery algorithm. Therefore, considering CS in target counting, we need to address the following questions:

1. How to design Ψ and Φ such that $A = \Phi\Psi$ satisfies the RIP property?
2. How to design efficient CS recovery algorithms for target counting?

6.2 Target Counting via Spatial Sparsity

The work in [36] represents one of the early effort in applying the CS theory to sparse target counting in sensor networks. In this paper, the authors partition the monitored area into N grids such that only $K \ll N$ of them contain targets. The target locations are modeled as a sparse signal $\Theta = [\Theta_1, \Theta_2, \ldots, \Theta_N]$, in which the indices of the nonzero elements represent the locations (grids) of the targets and the values of the nonzero elements represent the energy amplitudes of the targets at the corresponding grids they reside. Let X_i record the target energies measured by the ith sensor deployed in the monitored area. The sparse signal (Θ) is then linearly related to the sensor readings (X) through a sensing matrix Ψ, that is, $X = \Psi\Theta$. To define Ψ, a target energy decay model that takes into account the physics of the signal propagation and multipath effects is adopted. For simplicity, we denote by f_{ij} the decay model of the signal emitted by the target at location i and measured at location j.

Note that the definition of f_{ij} in [36] is quite complicated as the original design in [36] considers the case where a sensor at j measures L samples of the signal emitted by the target at location i. Therefore f_{ij} is a function of the sampling rate, the number of samples taken, the distance from the target to the sensor, the signal propagation speed, and the propagation attenuation constant. As the purpose of this chapter is to summarize the central idea of the major works in target counting, we focus on the simple case when $L = 1$ first. Thus our presentation in the following is a simplified version of the original work proposed in [36].

Assume that we have P sensors. Then the $P \times N$ target decay matrix Ψ can be defined as below:

$$\Psi = [\Psi_1^T, \Psi_2^T, \ldots, \Psi_P^T]^T = \begin{pmatrix} f_{11} & f_{21} & \cdots & f_{N1} \\ f_{12} & f_{22} & \cdots & f_{N2} \\ \vdots & \vdots & \vdots & \vdots \\ f_{1P} & f_{2P} & \cdots & f_{NP} \end{pmatrix} \tag{34}$$

Let $Y = \Phi X = \Phi\Psi\Theta$, where Φ is a $M \times P$ Gaussian random matrix with i.i.d. Gaussian-distributed entries, which satisfies the RIP property with any fixed basis. Then the authors can reconstruct Θ from Y through ℓ_1-minimization.

When $L > 1$, each Ψ_i is then defined as below:

$$\Psi_i = \begin{pmatrix} (f_{1i})_{l=1} & (f_{2i})_{l=1} & \cdots & (f_{Ni})_{l=1} \\ (f_{1i})_{l=2} & (f_{2i})_{l=2} & \cdots & (f_{Ni})_{l=2} \\ \vdots & \vdots & \vdots & \vdots \\ (f_{1i})_{l=L} & (f_{2i})_{l=L} & \cdots & (f_{Ni})_{l=L} \end{pmatrix} \tag{35}$$

In such a case, each X_i is a vector of length L. To conserve communication overhead, M random projections of each X_i will be transmitted to the central server for target signal recovery, where $M = O(K \log(N/K))$ is far less than X_i's original length L.

This method reduces the amount of communications significantly. Furthermore, since the CS reconstruction is robust against additive noise, the proposed method also shows a good performance under noisy conditions. Moreover, since the base target energy is assumed to be a variable that should be computed from sparse recovery (the values of the non-zero Θ entries), the proposed algorithm can be applicable to the case when the target base energies vary. This also implies that the algorithm can not count accurately when a grid may contain multiple targets, resulting the under-counting problem.

6.3 Target Counting via Bayesian Detection

The authors in [38] propose their own CS recovery algorithm for target counting. Similar to the work in [36], a N-grid monitored area is considered and the sparse vector $\Theta = [\Theta_1, \Theta_2, \ldots, \Theta_N]$ is a binary vector, with $\Theta_i = 1$ denoting that there exist targets in the ith grid, and $\Theta_i = 0$ otherwise. This work adopts a $N \times N$ sensing matrix Ψ following a target energy decay model defined as follows:

$$f_{ij} = \frac{E_t}{d_{ij}^{\alpha}} |G_{ij}| \tag{36}$$

where $|G_{ij}|$ captures the absolute value of the Raleigh fading of the target signal, d_{ij} is the distance between i and j, and E_t is the target base energy, i.e., the target energy measured at the target position. The sensing matrix Ψ is then defined as:

$$\Psi = \begin{pmatrix} f_{11} & f_{21} & \cdots & f_{N1} \\ f_{12} & f_{22} & \cdots & f_{N2} \\ \vdots & \vdots & \vdots & \vdots \\ f_{1N} & f_{2N} & \cdots & f_{NN} \end{pmatrix} \tag{37}$$

where f_{ij} denotes the signal energy at grid j for the targets at grid i. Instead of placing one sensor at each grid, this method randomly deploys M sensors at N grids, where $K < M \ll N$. To represent the sensor locations, a $M \times N$ matrix $\Phi = [\Phi_1, \Phi_2, \ldots, \Phi_M]^T$ is defined, where $\Phi_i = [\phi_{i1}, \phi_{i2}, \ldots, \phi_{iN}]$ is a row vector denoting the location of the ith sensor, i.e., $\phi_{ij} = 1$ if and only if the ith sensor resides

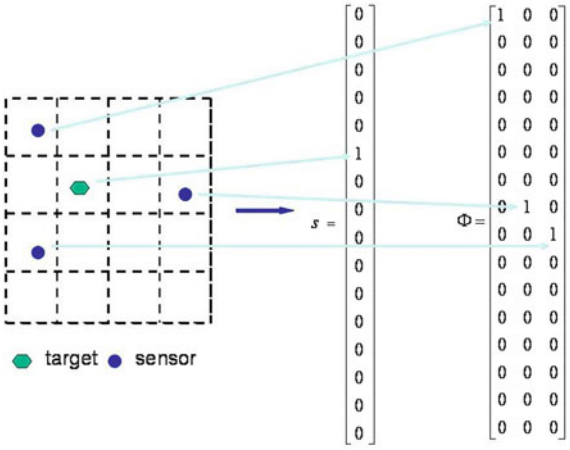


Fig. 8 Bayesian compressive sensing for target counting

at the jth grid (Fig. 8). By assuming A follows the RIP property, the authors claim that it can reconstruct Θ from $Y = \Phi\Psi\Theta$ according to the CS theory. Rather than using the conventional CS recovery algorithms, this work proposes a Bayesian recovery algorithm to recover Θ by taking advantage of the binary property of the sparse vector Θ. In the proposed algorithm, the conditional density function of Y over A and σ, and Θ over α are respectively expressed as:

$$P(Y|A, \sigma^2) = (2\pi\sigma^2)^{-M/2} exp(\frac{-1}{2\sigma^2}||Y - A\Theta||^2) \tag{38}$$

$$P(\Theta|\alpha) = (2\pi)^{-N/2} \prod_{n=1}^{N} \alpha_n^{1/2} exp(\frac{-\alpha_n \Theta_n^2}{2}) \tag{39}$$

where α is a vector of independent hyper-parameters that would be estimated later. The posterior distribution over the signal Θ can be obtained using the Bayes rule:

$$P(\Theta|Y, \alpha, \sigma^2) = \frac{P(Y|\Theta, \sigma^2)P(\Theta|\alpha)}{P(Y|\alpha, \sigma^2)} \tag{40}$$

which is actually a Gaussian distribution $N(\mu, \Sigma)$ with $\Sigma = (D + \sigma^{-2}\Psi^T\Psi)^{-1}$ and $\mu = \sigma^{-2}\Sigma\Psi x$, where $D = diag(\alpha_1, \alpha_2, \ldots, \alpha_N)$. The unknown parameters α and σ^2 can be estimated using the type-II maximum likelihood procedure [40]. Finally, this method recovers Θ by the posterior distribution over the signal conditioned on $\hat{\alpha}$ and $\hat{\sigma^2}$.

Simulation results indicate that compared with the conventional ℓ_1-minimization algorithm, the proposed Bayesian signal recovery algorithm achieves a comparable performance with an even smaller M. However, it still leaves many problems to be

discussed. First, the authors assume that its matrix A obeys the RIP without providing a rigorous proof. Second, due to the binary assumption of Θ, the proposed algorithm can only tell which grids contain targets but is not able to give an accurate count, especially when each grid may contain multiple targets. Moreover, the target decay model adopted by this approach assumes the availability of the base target energy, which may not be always available, rending the approach inapplicable if target base energies vary. In addition, the performance of the proposed method degrades quickly when the environment noise increases because the noise may disturb the assumed probabilistic distribution of the proposed model.

6.4 Target Counting via Orthogonal Matrix

Different from [38], where A is assumed to follow the RIP property, the authors in [37] introduce a preprocessing step to enforce A to obey the RIP. They define a sparse $N \times K$ matrix $\Theta = [\Theta_1, \Theta_2, \ldots, \Theta_K]$ to represent K targets' locations. Here Θ_k is a $N \times 1$ location vector with all elements equal to zero except $\Theta_k(n) = 1$, where n is the index of the grid point at which the kth target is located. This work adopts a target energy model slightly different from (36) without considering the Raleigh fading effect on signal propagation:

$$f_{ij} = \frac{E_t}{d_{ij}^{\alpha}} \tag{41}$$

A $N \times N$ matrix Ψ is then defined based on the above target decay model:

$$\Psi = \begin{pmatrix} f_{11} & f_{21} & \cdots & f_{N1} \\ f_{12} & f_{22} & \cdots & f_{N2} \\ \vdots & \vdots & \vdots & \vdots \\ f_{1N} & f_{2N} & \cdots & f_{NN} \end{pmatrix} \tag{42}$$

where f_{ij} denotes the signal energy at grid j for the targets at grid i. The same measurement matrix Φ proposed in [38] is adopted. Rather than simply assuming that $A = \Phi\Psi$ follows the RIP, the authors introduce a preprocessing step in order to obtain an RIP-compliant matrix:

$$Y' = QA^{\dagger}Y \tag{43}$$

where $Q = orth(A^T)^T$, and $orth(A^T)$ is an orthogonal basis for the range of A^T that obeys the RIP. The new CS measurement Y' can be further written as $Y' = Q\Theta + QA^{\dagger}\epsilon$. Since Q obeys the RIP, Θ can be recovered from Y' via ℓ_1-minimization. This method also introduces a post-processing procedure in order to compensate for the errors induced by the grid assumption. The proposed approach picks the dominant

coefficients in Θ_i whose values are above a certain threshold λ, and takes the centroid of these grid points as the location indicator of the targets.

Evaluations demonstrate that the proposed preprocessing step is able to generate RIP-compliant matrices. Moreover, the signal recovery results verify the reliability and robustness of the proposed approach. However, the required pre-processing step is computationally intensive and it does not always lead to convergence in ℓ_1-minimization [41].

6.5 Target Counting via Greedy Matching Pursuit (GMP)

The work by [39] for the first time rigorously justifies the validity of the applicability of CS on target counting by proving that the matrix A obeys the RIP property. It also proposes a compressive sensing theory-based problem formulation to count the targets from multiple different categories. In other words, the targets with different energy decay models and/or different base energy can be counted based on the new problem formulation.

The approach proposed in [39] considers a N-grid monitored area and defines a sparse vector $\Theta = [\Theta_1, \Theta_2, \ldots, \Theta_N]$, where $\Theta_i \in \{0, 1, 2, ..., m\}$ represents the number of targets in grid i and m is an integer indicating the largest possible number of targets a grid can hold. The target energy decay model is defined as follows:

$$f_{ij} = \frac{E_t}{d_{ij}^{\alpha}} G_{ij} \tag{44}$$

where G_{ij} is a complex random variable capturing the features of Raleigh fading with both its real and imaginary components following an independent and identical Gaussian distribution with a zero mean and a variance of σ^2. The matrices Ψ and Φ are defined similarly as those in Bayesian targeting counting [38]. In order to justify the applicability of CS to the target counting problem, the authors prove that $A = \Phi\Psi$ holds the RIP property with an overwhelming probability. It rewrites the CS measurement $Y = A\Theta$ as:

$$Y = A_r\Theta + A_i\Theta, \tag{45}$$

where A_r and A_i are the corresponding real components and imaginary components of the elements in A. Notice that if both A_r and A_i obey RIP, A must obey RIP. Since the real component and the imaginary component of A_{ij} are both independently and identically distributed Gaussian variables, it is sufficient to prove that A holds RIP when A_r holds. Based on probability theory, a rigorous proof is made in [39] to show that when $M \geq Klog(N/K)$, the probability for A_r to satisfy RIP tends to be 1.

Since Θ_i is drawn from a finite set $\{0, 1, \ldots, m\}$, instead of using conventional CS recovery algorithms, the authors propose a greedy matching pursuit algorithm

whose idea is to enumerate all possible values of Θ_i for all grids and find the one that contributes the most to the observation vector Y. This iterative algorithm can be sketched as follows. Let Y' be the residual target energy vector. Initially Y' is set to Y. At each step, the algorithm identifies the grid (i) and the number of targets at the grid that can maximize Az, where z is a $N \times 1$ vector containing 0 at z_j for all $j \neq i$. Then Az is subtracted from Y', indicating that Y' captures the remaining observed target energy when the grids with more number of targets are removed from the previous steps. The algorithm terminates when no grid that contains at least one target is found.

The authors further extend the proposed algorithm to multi-categorical target scenarios, which has never been addressed. Since the matrix Ψ is determined by the target energy decay model and each target category has its own energy decay property, for different targets, Ψ may be different. Assume that there are t categories of targets, with each having its own matrix Ψ characterizing the category-specific target energy dissipation features. Denote these matrices by Ψ_i for $i = 1, 2, \ldots, t$. Then the matrix Ψ_{multi} for the targets from multiple categories can be defined as follows:

$$\Psi_{multi} = \begin{pmatrix} \Psi_1 & 0 & \cdots & 0 \\ 0 & \Psi_2 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & \Psi_t \end{pmatrix} \tag{46}$$

Similarly, Φ_{multi} can be defined as:

$$\Phi_{multi} = \{\Phi_1, \Phi_2, \ldots, \Phi_t\}, \Phi_i = \Phi \text{ for } i \in 1, 2, \ldots, t \tag{47}$$

where Φ_i is the measurement matrix for category i. For the case when the sensors have the capability to measure the superposition of the target signals from different categories. The unknown vector containing the count and positioning information is denoted by $\Theta_t^c = \{\Theta_{t\,1}^c, \Theta_{t\,2}^c, \ldots, \Theta_{t\,t}^c\}^T$, where $\Theta_{t\,1}^c$ is a $N \times 1$ vector that denotes the location and number of targets of category i in the N grids. Let $A = \Phi_{multi}\Psi_{multi}$. Then we have $Y = A\Theta_t^c$, where Θ_t^c can also be recovered by the proposed GMP algorithm described above.

Compared with other works, the proposed method has the following advantages. First, it provides a rigorous proof for the applicability of CS theory to target counting, which guarantees that the signal can be recovered with a high probability. Second, besides target locations, the proposed algorithm can precisely count the targets in each grid while other works are only able to count roughly by telling the number of grids that contain targets. Third, the proposed method is able to differentiate targets from multiple categories, which has never been addressed before. In addition, the proposed algorithm shows a more stable performance while other CS algorithms fail to converge at a higher probability.

7 Discussion

In this section, we provide a discussion on the applicable environments of the different methods introduced in this book chapter. We consider different settings of the wireless sensor network and infer the influence of these settings on the performance of different counting methods. The settings under our consideration include: target characteristics, target deployment, sensor deployment, and the communication overhead. We summarize the different signal attenuation models utilized in existing works and their influences on the counting accuracy at the end of this section.

7.1 Target Characteristics

Generally speaking, there are two kinds of targets: the visually distinguishable targets and the visually indistinguishable targets. A photo-electric sensor can detect the presence of visually distinguishable targets by the visual contours, and also count the number of targets within the sensing region with high precision. In general cases, the external factors have little influence on the reading of a numeric sensor. As a result, the sensors' measurements tend to be stable when the number of target within the sensing region remains stable. The numeric counting model such as [14, 24] can provide a precise target count estimation based on the local observations in most cases. However, the working mechanism of numeric sensors is much more complex than that of energy sensors or energy-based binary sensors. For example, to extract the target count from local observations, pattern recognition process is executed in each numeric sensor. As a result, the manufacturing and maintenance costs of numeric sensors are higher than those of the energy sensors.

Note that energy counting and compressive counting models rely on the energy sensors. Compared with the numeric sensors, an energy sensor's reading is less stable due to the fact that the reading is related to not only the number of targets within its sensing region but also the relative position of the targets to the sensors. Due to the rapid attenuation effect of the target signal, the reading of an energy sensor may vary significantly even when the target position actually slightly changes. This will lead to unstable target count estimation for most methods. Although the energy counting model has these shortcomings, it is still the most powerful model in wireless sensor networks. First, no matter whether a target has a distinguishable contour or not, it always emits certain kinds of energy, i.e., heat, acoustic, or light, which can be measured by energy sensors. Second, the measurement of an energy sensor can not only reflect the presence of targets and the count of targets, but also the relative position of the targets. Combined with certain localization algorithms, position information can be inferred from the sensors' readings.

For the binary counting model, the sensor measurement may be obtained from both target contour information and energy information. Although this model is stable

and noise-resistant, its counting accuracy is the worst among all models. Actually, it only provides the lower bound of the target count in most cases.

7.2 Sensor Deployment

It has been pointed out that the counting accuracy is related to the sensor density and deployment patterns. There are a few works that focus on how to distribute the sensors appropriately to monitor the targets more effectively [42–46]. However, it is widely accepted that a uniform pattern can achieve the best counting performance in general cases.

Considering the sensor density, no conclusive work has been proposed. We summarize the influence of the sensor density by studying the reported simulation results, then we draw reasonable conclusions as follows.

For numeric counting and binary counting models, the optimal sensor density is achieved when the monitored area is fully covered with minimal overlapping. In such a case, the overlapping influence of the targets on different sensors is minimal. Oppositely, when sensors are densely deployed, the overlapping area becomes larger, and the estimation process becomes more complex. Consequently, the counting precision may not be improved.

For the energy counting model, the influence of the sensor deployment may vary. The DAM protocol counts targets by enumerating the sensors that have the local maximum energy readings. In this case, more sensors may help to differentiate signals from different targets. For other methods, the target count is estimated based on statistical modeling and approximation; thus the counting accuracy does not increase significantly after the sensor density reaches a certain level.

For compressive counting, the required number of sensors is limited by the minimum required signal sampling rate. More sensors may not help to improve the counting precision significantly, but may result in information redundancy.

7.3 Target Deployment

Both the target density and target distribution may influence the counting precision. There are five different target distributions that have been studied in the literature: uniform, hot-spot, multi hot-spot, piecewise uniform, and mixed.

Most algorithms are based on the assumption that targets are uniformly distributed in the monitored area. If this assumption does not hold, the counting accuracy decreases. There are a few works that consider the target counting problem in more general target distribution scenarios. In [14, 28], the authors model the distribution of the non-uniform targets with a piecewise uniform distribution. In [24], the authors take a regression model to recover the original target distribution. These algorithms

should be firstly considered in the case when the targets show apparently non-uniform patterns.

With regard to the target density, binary counting model can only be used for sparse targets. When the number of targets reaches a certain value, all sensors may simply output a value "1", which indicates that there is no way for the binary counting model to tell whether there are more targets added to the monitored area after the count of the targets reaches a certain number. For compressive counting model, the signal recovery algorithms are based on the assumption that the signals are sparse. As a result, compressive counting model is also mainly used for sparse targets. For energy counting model, the performance of algorithms based on signal separation may decrease after the count of targets reaches a certain level, since the signals from different targets may not be separated clearly in this case. However, for the algorithms based on signal landscape recovery and energy volume estimation, the density of the targets does not have significant influence on the counting accuracy. For the numeric counting model, the target density has little influence on the counting accuracy. Because the algorithms for this model are based on local observations, as long as the local measurements are precise, the final results generated by aggregating those local measurements should be precise. However, the computation cost may become unacceptable for [14] when the target count becomes large.

7.4 Communication Cost

The communication cost of target counting in a wireless sensor network is mainly related to the following two factors:

1. the amount of information transmitted;
2. the distance of information transmitted.

The binary counting model is the most economic in terms of the amount of information per communication since it only transmits 1 bit for each communication. For other models, the sensor measurement contains more bits.

Considering the second factor, DAM and the algorithms based on DAM can be considered as completely distributed, since they only involve local data communications. All other methods require sensors to transmit their states to a centralized server to make the final global decision.

Compressive counting can also be considered as a communication economic model. Although it is not distributed, and the information transmitted may be complex, it saves communication overhead by deploying a smaller number of sensors.

A brief summary in the applicability of the four models is listed in Table 1:

Table 1 Applicability of different counting model

Counting model	Measurement	Sensor deployment	Target deployment	Communication cost
Binary	Presence of targets	Depends	Sparse	Small
Numeric	Count of targets	Minimal overlap	General	Large
Energy	Energy	Depends	Depends	Depends
CS	Energy	Sparse	Sparse	Small

7.5 Energy Attenuation Model

For energy counting and compressive counting models, target signals are assumed to follow a specific attenuation model. In literature, there are mainly two models that are widely used, which are listed as follows:

1. $f_{\mathcal{E}}(d) = \frac{E_t}{(1+d)^{\alpha}}$
2. $f_{\mathcal{E}}(d) = \frac{E_t}{(d)^{\alpha}}$

where d represents the Euclidean distance from the signal source to a location, E_t denotes the base energy level of a target, and α is the attenuation factor. The difference of the two models lies in the denominator of the two equations. In the first model, an "1" is added to make the signal amplitude meaningful at the targets' location. For the second model, people usually takes the target's base energy level as the signal amplitude at the target's location. This is the only difference between the two attenuation models. Since there is no solid result from physics that can assert the validity of the two models, they need to co-exist. Even though the methods based on energy sensors may only consider one model, the counting accuracy should not be significantly different when the other model is considered.

8 Conclusion

In this book chapter, we provide a comprehensive survey over most target counting algorithms in wireless sensor networks. We categorize the state-of-art target counting algorithms into four categories, namely binary counting, numerical counting, energy counting, and compressive counting, based on the sensor's sensing capabilities and the underlying theoretical foundation. Technical details, pros and cons of these algorithms are reviewed. Comparisons and analysis are presented and the factors that may influence the counting precision are also discussed.

Future work in target counting may address the following problems.

1. In wireless sensor networks, sensor failures may be common. Therefore, fault tolerant target counting algorithms should be designed to overcome the problems caused by malfunctioning sensors.

2. Most target counting methods are centralized, which may involve intensive communication cost. However, distributed algorithms are preferred considering that the power supply of a sensor is very limited. Therefore algorithms with comparable counting precision and a less communication overhead are preferred.
3. It is possible to extend CS-based counting method to other sensing models such as numeric sensing models. In this case, the signal prototype can be different from those in energy sensing models. New assumptions should be made and proof of RIP under the new model is needed.
4. Adaptive nested model is another possible direction that deserves some attentions. A lot of works have been explored in target counting and the pros and cons of these works are basically clear. In this case, it is possible to employ a nested sensor network with different types of sensors. Multiple algorithms can be embedded in the network. The network detects the pattern of the targets and chooses the most appropriate method to estimate the count of targets adaptively.
5. The application of the target counting algorithms should be tested on real-world sensor network settings or in sensor network testbeds, while major existing works verify their design mainly based on simulation study.

References

1. M. Ding, D. Chen, A. Thaeler, X. Cheng, Fault-tolerant target detection in sensor networks, in *IEEE Wireless Communication and Networking Conference (WCNC)*, 2005
2. K. Xing, F. Liu, X. Cheng, D.H.-C. Du, Realtime detection of clone attacks in wireless sensor networks, in *The 28th International Conference on Distributed Computing Systems (ICDCS 2008)*, 2008, pp. 3–10
3. M. Ding, X. Cheng, Robust event boundary detection in sensor networks—a mixture model based approach, in *The 28th IEEE Conference on Computer Communications Mini-Conference*, 2009, pp. 2991–2995
4. S. Ren, Q. Li, H. Wang, X. Chen, X. Zhang, Analyzing object detection quality under probabilistic coverage in sensor networks, in *Thirteenth International Workshop on Quality of Service (IWQoS)*, 2005, pp. 107–122
5. S. Ren, Q. Li, H. Wang, X. Chen, X. Zhang, Design and analysis of sensing scheduling algorithms under partial coverage for object detection in sensor networks, in *IEEE Transactions on Parallel and Distributed Systems*, vol. 18, no. 3, pp. 334–350, 2007
6. M. Karakaya, H. Qi, Target detection and counting using a progressive certainty map in distributed visual sensor networks, in *Third ACM/IEEE International Conference on Distributed Smart Cameras, ICDSC 2009*, pp. 1–8 (2009)
7. R.G. Yuliy Baryshnikov, Target enumeration via euler characteristic integrals. SIAM J. Appl. Math. **70**(3), 825–844 (2009)
8. X. Cheng, A. Thaeler, G. Xue, D. Chen, TPS: A time-based positioning scheme for outdoor wireless sensor networks, in *INFOCOM 2004*, 2004, pp. 2685–2696
9. A. Thaeler, M. Ding, X. Cheng, iTPS: An improved location discovery scheme for sensor networks with long range beacons. J. Parallel Distrib. Comput. **65**(2), 98–106 (2005)
10. W. Wu, X. Cheng, M. Ding, K. Xing, F. Liu, P. Deng, Localized outlying and boundary data detection in sensor networks. IEEE Trans. Knowl. Data Eng. **19**(8), 1145–1157 (2007)
11. X. Cheng, H. Shu, Q. Liang, D.H.-C. Du, Silent positioning in underwater acoustic sensor networks. IEEE Trans. Veh. Technol. **57**(3), 1756–1766 (2008)

12. W. Cheng, A. Y. Teymorian, L. Ma, X. Cheng, X. Lu, Z. Lu, Underwater localization in sparse 3d acoustic sensor networks, in *The 27th IEEE Conference on Computer Communications (INFOCOM 2008)*, 2008, pp. 798–806
13. A.Y. Teymorian, W. Cheng, L. Ma, X. Cheng, X. Lu, Z. Lu, 3d underwater sensor network localization. IEEE Trans. Mob. Comput. **8**(12), 1610–1621 (2009)
14. S. Guo, T. He, M. Mokbel, J. Stankovic, T. Abdelzaher, On accurate and efficient statistical counting in sensor-based surveillance systems. Pervasive Mob. Comput. **6**(1), 74–92 (2010)
15. D. Li, K. Wong, Y.H. Hu, A. Sayeed, Detection, classification, and tracking of targets. IEEE Signal Process. Mag. **19**(2), 17–29 (2002)
16. Q. Fang, F. Zhao, L. Guibas, Counting targets: Building and managing aggregates in wireless sensor networks. Tech. Rep, in *Palo Alto Research Center Technical Report*, 2002
17. J. Chen, K. Cao, K. Li, Y. Sun, Distributed sensor activation algorithm for target tracking with binary sensor networks. Cluster Comput. **14**, 55–64 (2011)
18. Z. Wang, E. Bulut, B. Szymanski, Distributed target tracking with directional binary sensor networks, in *IEEE Global Telecommunications Conference, GLOBECOM 2009*, pp. 1–6 (2009)
19. W. Kim, K. Mechitov, J.-Y. Choi, S. Ham, On target tracking with binary proximity sensors, in *Proceedings of the 4th international symposium on Information processing in sensor networks*, ser. IPSN '05, 2005
20. N. Shrivastava, R. Mudumbai, U. Madhow, S. Suri, Target tracking with binary proximity sensors. ACM Trans. Sen. Netw. **5**, 30:1–30:33 (2009)
21. M. Zhu, S. Ding, Q. Wu, R.R. Brooks, N.S.V. Rao, S.S. Iyengar, Fusion of threshold rules for target detection in wireless sensor networks. ACM Trans. Sen. Netw. **6**, 18:1–18:7 (2010)
22. N. Shrivastava, R.M.U. Madhow, S. Suri, Target tracking with binary proximity sensors: fundamental limits, minimal descriptions, and algorithms, in *Proceedings of the 4th international conference on Embedded networked sensor systems*, 2006, pp. 251–264
23. S. Gandhi, R. Kumar, S. Suri, Target counting under minimal sensing: Complexity and approximations, in *Proceddings of the 4th international workshop Algosensors*, 2008, pp. 30–42
24. D. Wu, D. Chen, K. Xing, X. Cheng, A statistic approach for target counting in sensor-based surveillance systems, in *The 31th IEEE Conference on Computer Communications (INFOCOM 2012)*, 2012
25. K. Levenberg, A method for the solution of certain non-linear problems in least squares. Q. Appl. Math. **2**, 64–68 (1994)
26. W. Hardle, *Applied Nonparametric Regression*, 1st edn. (Cambridge University Press, Cambridge, 1992)
27. P.J. Bickel, K.A. Doksum, *Mathematical Statistics Basic Ideas and Selected Topics*, vol. 1, 2nd edn. (Prentice Hall, Englewood Cliffs, 2001)
28. D. Wu, X. Cheng, D. Chen, W. Cheng, B. Chen, W. Zhao, A monte carlo method for target counting, in *Proceedings of the 31th IEEE international conference on Distributed Computing in Sensor Systems*, 2011, pp. 750–758
29. Y. Guo, B. Hua, L. Yue, Energy-based target numeration in wireless sensor networks, in *FGCN '07: Proceedings of the Future Generation Communication and Networking*, 2007, pp. 380–385
30. Q. Fang, F. Zhao, L. Guibas, Lightweight sensing and communication protocols for target enumeration and aggregation, in *Proceedings of the 4th ACM International Symposium on Mobile ad hoc networking and, computing*, 2003, pp. 165–176
31. A. Franz, Voronoi diagramsa survey of a fundamental geometric data structure. ACM Comput. Surv. **23**, 345–405 (1991)
32. N. Metropolis, S. Ulam, The monte carlo method. J. Am. Stat. Assoc. **5544**, 335–341 (1949)
33. C. Robert, G. Casella, *Monte Carlo Statistical Methods*, 2nd edn. (Springer, New York, 2004)
34. E. Cands, M. Wakin, An introduction to compressive sampling. IEEE Signal Process. Mag. **25**(2), 21–30 (2008)
35. D. Needell, R. Vershynin, Signal recovery from incomplete and inaccurate measurements via regularized orthogonal matching pursuit. IEEE J. Sel. Top. Sig. Process. **4**(2), 310–317 (2010)
36. V. Cevher, M.F. Duarte, R.G. Baraniuk, Distributed target localization via spatial sparsity, in *EUSIPCO*, 2008, pp. 25–29

37. C. Feng, S. Valaee, Z. Tan, Multiple target localization using compressive sensing, in *GLOBECOM'09: Proceedings of the 28th IEEE conference on Global, telecommunications*, 2009, pp. 4356–4361
38. J. Meng, H. Li, Z. Han, Sparse event detection in wireless sensor networks using compressive sensing, in *The 43rd Annual Conference on Information Sciences and Systems (CISS)*, 2009, pp. 181–185
39. B. Zhang, X. Cheng, N. Zhang, Y. Cui, Y. Li, Q. Liang, Sparse target counting and localization in sensor networks based on compressive sensing, in *Proceedings of IEEE INFOCOM*, pp. 2255–2263 (2011)
40. D.J. MacKay, A practical bayesian framework for backprop networks. Neural Comput. **4**, 448–472 (1991)
41. Q. Cui, J. Deng, X. Zhang, Compressive sensing based wireless localization with minor component analysis. Beijing University of Posts and Telecommunications, Tech. Rep., 2012
42. L. Lazos, R. Poovendran, J. Ritcey, On the deployment of heterogeneous sensor networks for detection of mobile targets, in *5th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks and Workshops, 2007*. WiOpt 2007, Apr 2007, pp. 1–10
43. X. Cheng, D.-Z. Du, L. Wang, B. Xu, Relay sensor placement in wireless sensor networks. Wireless Netw. **14**(3), 347–355 (2008)
44. N. Akshay, M. Kumar, B. Harish, S. Dhanorkar, An efficient approach for sensor deployments in wireless sensor network, in *International Conference on Emerging Trends in Robotics and Communication Technologies (INTERACT 2010)*, pp. 350–355 (2010)
45. M. Cardei, Y. Yang, J. Wu, Non-uniform sensor deployment in mobile wireless sensor networks, in *International Symposium on a World of Wireless, Mobile and Multimedia Networks, 2008. WoWMoM 2008*, 2008, pp. 1–8
46. X. Bai, Z. Yun, D. Xuan, T. Lai, W. Jia, Deploying four-connectivity and full-coverage wireless sensor networks, in *The 27th IEEE Conference on Computer Communications Mini-Conference*, 2008, pp. 296–300

Part IV
Coverage and Localization in Three-Dimensional Wireless Sensor Networks

Chapter 8
Coverage and Connectivity in 3D Wireless Sensor Networks

Usman Mansoor and Habib M. Ammari

Abstract A wireless sensor network (WSN) is categorized as three-dimensional (3D) when variation in the height of deployed sensor nodes is not negligible as compared to length and breadth of deployment field. The fundamental problem in such 3D networks is to find an optimal way to deploy sensor nodes needed to maintain full (or targeted degree of) coverage of monitored volume and reliable connectivity as desired by network designers. The solution should yield lower bound on number of nodes needed to achieve full coverage and connectivity. However, optimizing coverage and connectivity in 3D WSNs comes with its inherent complexities and intrinsic design challenges. 3D WSNs are not only difficult to visualize but their analysis is also computationally intensive. This literature summarizes major work conducted in the domain of coverage and connectivity in 3D WSNs. It studies different placement strategies, fundamental characteristics, modeling schemes, analytical methods, limiting factors, and practical constraints dealing with coverage and connectivity in 3D WSNs.

1 Introduction

Three-dimensional (3D) wireless sensor networks (WSNs) lie in the forefront of many advanced industrial applications. For example, airborne WSNs for supporting intelligent computer vision [18], helping overcome human paropsia [65], air borne defense systems [1], underwater monitoring including acoustic networks [46, 61], climate monitoring and weather forecasting, and many more. However, the complexity of 3D WSNs in terms of coverage and connectivity is prohibitively higher as opposed to two-dimensional (2D) networks. The 2D network model works fine with terrestrial networks where variation in height of network is small compared to

U. Mansoor · H. M. Ammari (✉)
WiSeMAN Research Lab, CIS Department, University of Michigan, Dearborn, MI, USA
e-mail: hammari@umd.umich.edu

H. M. Ammari (ed.), *The Art of Wireless Sensor Networks*,
Signals and Communication Technology, DOI: 10.1007/978-3-642-40066-7_8,

length and width of WSN Coverage Field. However, most airborne and underwater applications necessitate comprehensive understanding of 3D network design. The fundamental problem in such 3D networks is to find an optimal way to deploy sensor nodes needed to maintain full coverage of monitored volume and reliable connectivity as desired by network designers. The solution should yield low bound on number of nodes needed to achieve full coverage and connectivity.

1.1 Curse of Dimensionality

3D WSN model comes with its inherent complexities and intrinsic design challenges. 3D WSNs are not only difficult to visualize but their analysis is also computationally intensive. This makes self-organization and optimization schemes difficult to implement since most sensor nodes have limited computational capacity and life span. Generally speaking, 3D WSNs require higher node densities to achieve same degree of coverage as 2D WSNs. Furthermore, optimal 3D node deployment usually requires complex mathematical modeling and simulation. Poduri et al. [47] aptly names these design challenges as "Curse of Dimensionality," and is a good reference for introduction to intricacies of 3D WSNs. In this section we highlight these fundamental challenges (presented in [47]) while designing a 3D WSN network.

Poduri et al. [47] explains that for randomly deployed networks, the critical node density that results in almost surely connected giant component is greater in 3D than in 2D with. Therefore almost surely connected randomly deployed components will have highly dense topology compared to 2D environments [31, 47]. Similarly for coverage, to avoid coverage-less patches, the sensing spheres of the nodes have to maintain separation of less than Rs (sensing radius) since spheres do not tile in space. This results in considerably higher critical nodal density for 3D environments and requires careful planning and considerations for node placement strategies.

Apart from distance measurements, angular information between nodes or reference points can also be usefully employed for many algorithms (e.g., localization, directional beacons) and protocols. For example, in several power control techniques [17, 66], the angle between adjacent nodes is used to control transmission power thresholds. The computational complexity to handle angular information-based schemes and algorithms increases markedly for 3D compared to 2D environments.

3D WSNs also require considerably complex and computationally intensive analysis to insect degree of coverage of deployed sensing nodes in 3D field. Intersection of sensing regions is used to determine the degree of network coverage and detect any coverage-less patches. For 2D WSNs, sensing regions are modeled as circular discs and their intersection analysis takes computational order of degree $O(d^2)$, where d is the number of node neighborhood sensing region intersections. However, for 3D WSNs, the analysis becomes considerably more complicated $O(d^3)$. Furthermore, d for 3D WSNs is also higher compared to 2D WSNs [47]. This makes coverage planning a mathematically complicated and computationally intensive task. Furthermore,

for optimum regular deployment of nodes in 3D space, nodes are placed according to space filling regular polyhedron patterns such as octahedrons, cube etc. The analysis of space filling polyhedral for optimum packing based on desired degree of coverage and connectivity can be a complicated undertaking [47]. The scenario is further complicated due to tendency of most 3D environments (submerged, terrestrial 3D) being heavily affected by obstacles. The analysis of obstacles in terms of signal propagation, graph connectivity, coverage effect can be considerably more challenging for 3D environments compared to 2D WSNs, and might require dedicated software and analytical techniques to yield reliable and quick solutions.

1.2 Relevance of 3D Design

3D WSNs have found applications in many emerging applications including defense, environmental research, weather monitoring, surveillance, space research, communication, and exploration [3, 18, 29, 43, 46, 48, 57, 61, 65]. A WSN is categorized as 3D when the variation in height of the deployed sensors is not negligible as compared to length and width of the deployment field. Conventionally, sensor networks are usually visualized as 2D networks. This assumption is valid for most terrestrial scenarios since nodes are deployed in same planes and there is usually little or no across planes nodal communication. However, this 2D model loses its relevance for most submerged and airborne deployments. In fact it is shown in [40, 43, 75] that even terrestrial networks deployed on complex 3D surfaces need to be analyzed as 3D WSNs as opposed to 2D WSNs since the 3D model-based node placement strategies yield better coverage and reduced number of sensor nodes. Although as important and crucial as it may sound, 3D WSNs is still relatively unexplored niche in the WSN domain, and it has only been since recently that concentrated research efforts have been undertaken in this domain. Especially, Submerged WSNs commonly referred to as Under Water-Acoustic Sensor Networks (UW-ASNs) have received serious attention due to defense and economic potential of its prospective applications. For example, Ocean Sampling Networks consist of network of sensors and Odyssey Class AUVs (Autonomous Underwater Vehicles) which collectively can perform synoptic, cooperative adaptive sampling of 3D costal environments [13, 45]. UW-ASNs can be used in environmental monitoring (chemical, biological, nuclear) in rivers, lakes, oceans etc. [72]. UW-ASNs can also be used to explore underwater oilfields and other natural resources. Disaster prevention and early warning systems to warn against tsunami threats and seismic activities including seaquakes have huge applications for WSNs in general and 3D WSNs in particular [48, 49]. Assisted navigation can help improve safety and efficiency in navigation systems. Distributed tactical surveillance based on 3D WSNs detects and classifies submarines and small delivery vehicles (SDVs). Such systems are increasingly becoming indispensable part of any maritime defense systems. Airborne applications include but not limited to: airborne radars, scanners, climate monitoring, surveillance, drone technology, communication systems, and surface mapping [1, 20, 59].

Among many design challenges in 3D WSNs which include routing, energy efficiency, latency, remote configuration, practicality; coverage and connectivity lies at the forefront. It deals with optimal placement of nodes in 3D space to achieve targeted degree of coverage and connectivity with minimum number of sensor nodes.

This survey work summarizes the major work done in the field of coverage and connectivity in 3D Wireless Sensor Networks. Section 2 enlists basic taxonomy. Section 3 entails 3D cell partitioning based on space filling polyhedrons. Section 4 studies the transmission and sensing radii constraints for 3D WSNs. These fundamentals are essential to model and analyze any 3D WSN. Section 5 briefly touches the domain of Conditional Connectivity. Section 6 analyzes the lattice patterns for full coverage and *k*-connectivity. Section 7 shows use of Continuum Percolation to find critical node density to achieve specified degree of coverage and connectivity. Section 8 is dedicated to UW-ASNs and different deployment methods for submerged networks. Section 9 shows the relevance of 3D WSNs for complex terrestrial surfaces. Section 10 studies the Mobility in 3D WSNs using Virtual Forces Model. Section 11 summarizes some of the relevant works, and finally in Sect. 12 we address the research challenges and open problems in this domain.

2 Basic Taxonomy

This section enumerates essential terminologies for 3D WSNs.

Preliminaries: coverage and connectivity	Spherical sensing and communication ranges	In 3D environments, the sensor node coverage and communication zone are usually modeled as spheres with radii Rs and Rc centered at the node's location. Any point within the coverage sphere (<Rs) is assumed to be seen (monitored) by the node. Similarly the radius of communication sphere dictates the transmission range of the sensor node (<Rc). Typically the communication range of sensor node is greater than its sensing range (Rc > Rs)
	Coverage	A point in the 3D Field of Interest (FoI) is said to be 1-covered if it belongs to at least sensing sphere (lies within the sensing sphere) of one sensor, and k-covered if it belongs to at least sensing spheres of k sensors. Depending on the application an area may require that multiple sensors monitor each point in the field of interest
	Connectivity	Connectivity in WSN is often represented by a graph, with nodes making the vertices. Propagation conditions are usually modeled by transmission radius which specifies the communication range of each node. A graph is connected if there exists a path between any pair of its vertices. For the network to be declared connected, then any pair of nodes should be able to communicate with each other, possibly taking multi-hops through relay nodes

(continued)

	k-connectivity	A network is k-connected if there exists at least k disjoint paths between any two nodes. Higher the degree of connectivity, greater the network has resilience to node failure
	k-coverage	A 3D region is said to be k-covered if each point in 3D region is covered by at least k sensors. This constraint is known as k-coverage in which the k represents the number of nodes that watch each point. Requiring a k value of more than one will add complexity to the coverage algorithm
	Hop diameter	The hop diameter between two nodes u and v is the shortest path needed to traverse (in terms of hops) to reach v from u or vice versa
	Full coverage	When every point in 3D space is at least covered by the sensing sphere of one sensor
	Reduced coverage	Strict planning for k-coverage of 3D space, where k = 1. Maximum effort to avoid k>1 patches in the coverage region
	Maximum coverage	Planning for node placement for Rc/Rs resulting in maximum coverage of 3D space with minimum number of sensor nodes
Mathematical tools	Space filling polyhedron	Three-dimensional shape consisting of finite number of polygonal faces and surrounds a bounded volume in 3D space
	Voronoi tessellation	Decomposition of a metric space, determined by distances to a specified family of objects (subsets) in the space
	Kepler conjecture	Mathematical conjecture about sphere packing in 3D space: no arrangement of equally sized spheres filling space has a greater average density than that of the cubic close packing (face-centered cubic) and hexagonal close packing arrangements. Density is slightly greater than 74%
	Kelvin conjecture	What space-filling arrangement of similar cells of equal volume has minimal surface area? Proposed solution was a 14-sided truncated octahedron (TO) having a very slight curvature of the hexagonal faces. In 1994, space-filling unit cell consisting of six 14-sided polyhedra and two 12-sided polyhedra with irregular faces and only hexagonal faces remaining planar was discovered. This structure has an isoperimetric quotient of 0.765, or approximately 1.0% more than curved TO's cell
	Continuum percolation	Percolation occurs with positive probability if any given random shape is part of infinite clump of random shapes. There exists a node density for which network detectability reaches almost i.e. an event is detected with probability of almost 1

(continued)

	Volumetric quotient	The ratio between volumes of polyhedron unit cell and the corresponding circum-sphere, the sensing range of the node
Spherical models	Ball centered	Space inside the sphere. Open Ball does not include boundaries of sphere
	Sphere centered	Spherical sensing range with radius r centered at node location. When used along Ball centered, sphere usually only refers to boundaries of sphere will ball refers to space inside the boundaries of sphere
Conditional	Forbidden faulty set	A set of faulty sensors that includes the entire neighbor set of a given sensor
	Conditional connectivity	The minimum size of forbidden faulty set for which graph becomes disconnected.
k-connectivity and full coverage	Low connectivity and Full coverage	Planning for full coverage while maintaining k connectivity $k \leq 4$
	High connectivity and full coverage	Planning for full coverage while maintaining k connectivity $k = 6, 14$
Continuum percolation	Covered component	A covered component (or covered region) is a maximal set of sensing spheres (i.e., not included in any other subset except when it is equal to the original entire set of sensing spheres) whose corresponding sensors are collaborating directly or indirectly
	Connected component	A connected component is a maximal set of communication spheres whose corresponding sensors are communicating directly or indirectly
	Coordinated component	A coordinated component is a maximal set of concentric sensing and communication spheres whose corresponding sensors are coordinating directly or indirectly
Network	Homogeneous	Consists of sensor nodes with same abilities, such as computing power and sensing range, transmission range, node lifetime
	Heterogeneous	Consists of sensor nodes with different abilities, such as different computing power and sensing range, transmission range, node lifetime
	Hierarchical	Some nodes in the network may have different types, abilities or assigned roles e.g., Backbone nodes
	Non-hierarchical	All nodes in the network have same types, abilities and assigned roles

(continued)

UW-ASN	2D	Sensors at bottom of the Ocean
	3D	Sensors f loat at different depths
Deployment strategies	3D Random	Simplest Strategy, No coordination from surface station. Sensors are randomly deployed, sensors choose their depth randomly. Each sensor informs its position
	Bottom-random	Sensors are randomly deployed at the bottom. Each sensor informs its location. Surface station then calculates the optimal depth for each sensor. Sensors then take that depth
	Bottom-grid	Assisted by AUVs. Sensors deployed to pre-defined target locations to obtain grid deployment at the bottom of ocean. Each sensor then accordingly floats to its designated depth

(continued)

3 Dimensional Cell Partitioning

3D Cell partitioning lies at the core of different placement strategies. Cell partitioning scheme is usually dictated by the ratio of communication to sensing radius. A network designer wants to optimally divide the 3D space into 3D cells to achieve desired degree of connectivity and coverage while minimizing the number of required nodes. 3D Cell partitioning can be broadly classified into two categories.

1. Maximal Coverage (1-Coverage)
2. k-coverage ($k > 1$)

In 1-coverage, it is aimed to minimize the overlap between sensing spheres without having any coverage-less patches. This is usually achieved by dividing the 3D space into cell patterns based on space filling polyhedrons. The choice of space filling polyhedron is dependent on the communication to sensing radius of the WSN nodes.

k-coverage is the logical extension of 1-coverage. In k-coverage, by decreasing the distance between sensing nodes, desired sensing sphere overlap is achieved. Usually careful and computationally intensive analysis is required to optimally solve for k-coverage.

3.1 Volumetric Quotient-Based Approach

3.1.1 Preliminaries: Space Filling Polyhedron

A polyhedron is a three-dimensional shape consisting of finite number of polygonal faces and surrounds a bounded volume in 3D space. The straight lines where faces meet are edges, and edges meet at points called vertices. There are five regular

polyhedrons or Platonic Solids (polyhedrons having convex congruent faces): cube, dodecahedron, icosahedron, octahedron, and tetrahedron.

A space-filling polyhedron also known as plesiohedron can be used to generate tessellation of space. There are five space-filling convex polyhedral having regular faces: the triangular prism, hexagonal prism (HP), cube (CB), truncated octahedron (TO) [55, 70], and gyrobifastigium [41], with cube being the only Platonic Solid possessing this property [28, 32, 69]. Of all the 13 Archimedean solids (highly symmetric, semi regular convex polyhedron composed of two or more types of regular polygons meeting in identical vertices), only truncated octahedron tiles space [58]. The determination of maximal density arrangements for non-tiling polyhedrons is a highly computationally intensive and brutally difficult problem. The rhombic dodecahedron (RD), elongated dodecahedron, and squashed dodecahedron are also space-fillers. Combination of tetrahedrons and octahedrons also fills space. In addition octahedrons truncated octahedrons, and cubes, combined in the ratio 1:1:3 can also fill space [4, 69].

3.1.2 The Big Four (CB, HP, RD, TO)

The number of nodes required for 3D coverage is inversely proportional to volumetric quotient $V/(4\pi R^3/3)$ of space filling polyhedron. The problem simplifies to finding a space filling polyhedron with highest volumetric quotient. To fully cover a 3D space, each Voronoi cell must have maximal volume for given sensing range R. Neglecting boundary effect, total number of nodes required would be ratio of volume of 3D space to be covered to volume of one Voronoi cell. To achieve highest Voronoi cell volume, the radius of circumsphere must be equal to sensing range R.

Alam et al. [4, 6–8] makes use of Voronoi Tessellation of 3D space for creation of truncated octahedral cells for optimal coverage and connectivity. It uses Kelvin Conjecture for placement of nodes in the middle of truncated octahedrons. Alam et al. [4] proves that octahedron placement strategy is valid if the ratio of transmission to sensing range of nodes is at least 1.7889. Hexagonal Prism or Rhombic dodecahedron (Refer to Fig. 1 for illustrations) placement strategy needs to be adopted for transmission to sensing range ratio of 1.4142–1.7889. The use of truncated octahedron (TO) is justified by emphasizing the similarity of the problem to Kelvin's Conjecture.

Table 1 enlists the volumetric quotients of TO, CB, HP, RD. Since Kelvin Problem is essentially finding a space-filling polyhedron with minimal surface area to volume ratio, and sphere has the volumetric quotient of 1, such a polyhedron would ideally approximate sphere. The solution to such single cell shape Kelvin problem is truncated octahedron. The work in [4] also compares the truncated octahedron with rhombic dodecahedron, the hexagonal prism, and the cube along with their placement strategies to deploy nodes such that resultant Voronoi cells are chosen space filling polyhedrons. Jawahar et al. [38] present a similar work which proves the choice of truncated octahedron on basis of its highest Volumetric Quotient.

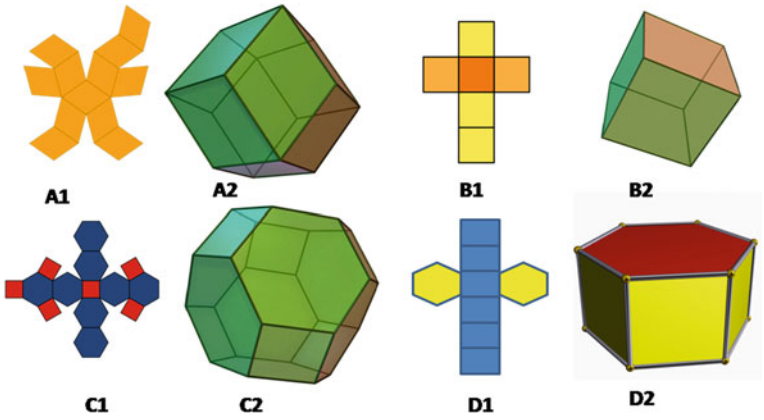


Fig. 1 Rhombic dodecahedron (A2), Cube (B2), Truncated octahedron (C2), Hexagonal prism (D2). A1,B1,C1,D1 are corresponding Net Images. (The illustrations are taken from http://mathworld.wolfram.com)

The required minimal transmission range to maintain connectivity among nodes depends on the choice of polyhedron. When cube is chosen the distance between two neighboring nodes is $2r_s/\sqrt{3}$. Therefore r_c/r_s should be $\geq 2/\sqrt{3}$. In HP $r_c/r_s \geq \sqrt{2}$ to maintain connectivity along x, y planes, and $r_c/r_s \geq 2/\sqrt{3}$ along z plane. The minimum of maximum ratio is selected to maintain connectivity in all the three planes [4, 8].

3.1.3 TO versus CB, HP, and RD

Interestingly, TO-based deployment strategy is not necessarily the best solution in every scenario. Although Alam et al. [6] proposes the use of truncated octahedron shaped cells for in 3D for deployment of sensor nodes, the ratio, minimum sensing radius to communication radius for TO, is also the greatest of all four. Simulation results in [6] show that truncated octahedron (TO)-based cell division strategy outperforms cube (CB), hexagonal prism (HP), and rhombic dodecahedron (RD) respectively but at the expense of larger sensing radius (considering communication radius to be constant). However, TO-based space division model proves its worth when the number of active nodes needed to achieve full coverage is observed. Usually a tradeoff needs to be achieved when considering sensing radius of each node and number of total nodes required. Table 2 compares the sensing to transmission radii required for all deployment schemes.

Jawahar et al. [38], Wang et al. [62], and Jiang et al. [39] also justify the use of truncated octahedron model on the basis of better energy dissipation characteristics of the network and prolonged network lifetime. Jawahar et al. [38] claims increased network lifetime by a factor of 19 for 3D WSNs. By dividing the densely deployed

Table 1 Volumetric quotient of space filling polyhedrons

Polyhedron	Cube (CB)	Hexagonal prism (HP)	Rhombic dodecahedron (RD)	Truncated octahedron (TO)
Volumetric quotient (VQ)	0.36755	0.477	0.477	0.68329
VQ formula	$a^3/\frac{4}{3}\pi\left[\frac{\sqrt{3}}{2}a\right]^3$	$\frac{3\sqrt{3}}{2}a^2h/\frac{4}{3}\pi\left[\sqrt{a^2+h^2/4}\right]^3$	$2a^3/\frac{4}{3}\pi a^3$	$8\sqrt{2a^3}/\frac{4}{3}\pi\left[\frac{1}{2}\sqrt{10}a\right]^3$
Definition of a	Length of side: a	Length of side a. optimal $h = a\sqrt{2}$	Each side of two original cubes: a	Length of each edge: a
No. of nodes needed compared to TO	5.9 % more	43.25 % more	43.25 % more	NA
Min transmission range	1.1547R	1.4142R	1.4142R	1.7889R

The data is summarized from [4]

Table 2 Comparison of different cell partitioning schemes

Polyhedron	Sensing radius	Number of active nodes required to achieve full coverage
Truncated octahedron (TO)	0.542326 × transmission radius	1×TO
Cube (CB)	0.5 × transmission radius	2.372239×TO
Hexagonal prism (HP)	0.53452 × transmission radius	1.82615×TO
Rhombic dodecahedron (RD)	0.5 × transmission radius	1.49468×TO

3D sensor field into a number of truncated octahedron cells, only one node in each cell is in active mode while rest is put in idle mode. In this manner connectivity and coverage is maintained by using minimal number of active cells. Refer to Fig. 2. Nodes in a cell keep swapping active-idle roles, resultantly increasing the overall network lifetime.

However, it has to be said there are some fundamental assumptions in this simulation work which might make this approach impractical in real-world scenarios. To state a few: the use of swapping active-idle roles and truncated octahedrons may result in reduced k-connectivity, increased number of nodes would be required for deployment, strict node deployment policies will need to be implemented along with very efficient network wide synchronization scheme will be required to achieve such a high factor in network lifetime increase.

Xiao et al. [71] gives simulation results of a design experiment. Though the experiment does not specifically follow any polyhedron-based node deployment strategy, the experiment is considerable similar in nature to [38]. In short, the experiment tends to show that degree of network connectivity and coverage increases as node spatial density increases.

Watfa et al. [63] uses body centered cubic (bcc) lattice (shown in Fig. 3) for 3D node placement strategy. It is explained in [63] that the thinnest cover of a region by spheres is obtained when the centers of the spheres are at the vertices of the bcc lattice. If the distance between adjacent vertices is one unit, then the entire region can be covered by the copies of a sphere whose covering radius is $\text{Rcover} = \sqrt{5}/2 = 1.1180$. Such a lattice is periodic and is completely reduced. Thus the deployment will be optimal if the spacing between the centers of adjacent sensor nodes equals Rs/1.1180. Table 3 shows the simulation results for covering a 3D field of dimensions $10\times10\times10$ units^3 and sensing radius of x units.

Commuri et al. [22] gives details of series of simulated design experiments which demonstrate the energy efficiency of bcc lattice-based deployment strategy compared to random deployment of sensor networks. The work entails different simulated scenarios. In each scenario it is shown that energy dissipated for a reduced coverage network—as referred in text—is less than a randomly deployed sensor network. A reduced coverage network is basically 1-coverage 3D WSN. Failure of a single node in reduced coverage network will result in loss of coverage in 3D WSN field. Since reduced coverage deployment strategy is based on bcc lattice pattern, the work in [22] also deduces that number of nodes required to cover a 3D field is less using bcc lattice-based deployment strategies as opposed to randomly deployed WSNs. Similar

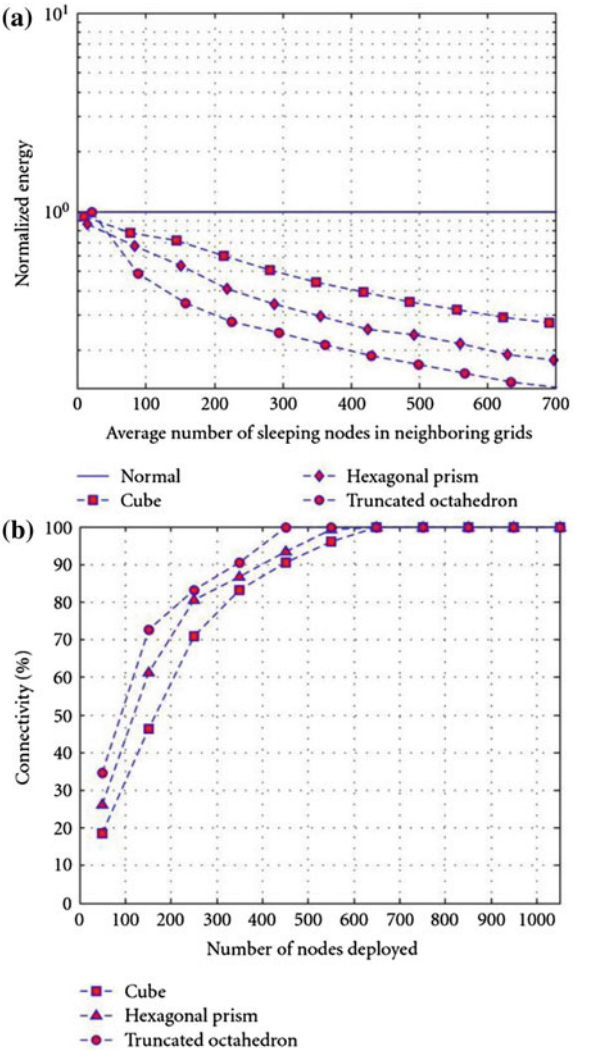


Fig. 2 **a** Shows the increased number of sleeping nodes for using *truncated octahedron* cells. **b** Shows that 100 % connectivity is achieved for lesser number of nodes when using truncated octahedron cell as opposed to *cube* or *hexagonal prism*. The figure is a snapshot of results achieved in simulation in [38]

to [6, 38], Commuri et al. [22] also puts the nodes in idle mode such that minimal number of nodes are active at given time resulting in reduced coverage (1-coverage) of the network. A self-healing mechanism is also proposed in which a failing node in a reduced network alerts its neighbors about the impending failure. Inactive nodes in the neighborhood are activated and hence loss of coverage is avoided. Watfa et al. [64] proposes a distributed algorithm for optimal coverage in 3D field. It is proposed that if the total overlap volume of a sensing sphere S_i, formed by the overlap of sensing spheres of neighbors of S_i, is less than the sensing sphere volume S_i, then S_i is not completely covered by its neighbors and hence it must always remain active.

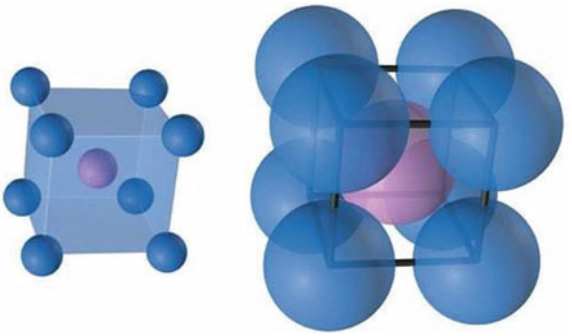


Fig. 3 Body centered cubic lattice

Table 3 Random deployment versus bcc lattice-based deployment scheme

Sensing radius	Random deployment	Bcc lattice deployment
2	182	54
1.5	325	128
1	773	396

Simulation results from [63] which indicate that lesser number of nodes are required to cover a field when using bcc lattice deployment as opposed to random deployment scheme. $10\times10\times10$ units^3 field is covered

3.2 Spherical Overlap Approach (k-Coverage)

3.2.1 Reuleaux Tetrahedron-Based Coverage Analysis

The reuleaux tetrahedron (also sometimes known as spherical tetrahedron) is three-dimensional solid formed by intersection of four spheres of radius r centered at the vertices of a regular tetrahedron with side length r (Refer to Fig. 6). The sphere through each vertex passes through the other three vertices, which also form vertices of the reuleaux tetrahedron. The reuleaux tetrahedron has the same face structure as a regular tetrahedron, but with curved faces: four vertices, and four curved faces, connected by six circular-arc edges [25, 68].

Ammari and Das in [9, 10] extend the work in [4] by proposing the use of Reuleaux tetrahedron to characterize k-coverage of a 3D field in WSNs and find the corresponding minimal sensor spatial density.

Ammari and Das in [11] exploited the geometric properties of reuleaux triangle to define set of conditions to fully K-cover a 2D sensing field. A reuleaux triangle of side r is formed by the intersection of three symmetric congruent discs. It has central regular triangle surrounded by three curved regions. Importantly, it has constant width equal to r. Refer to Fig. 4. Constant width is achieved by a curve in which the distance between two opposite parallel tangent lines to its boundary is the same,

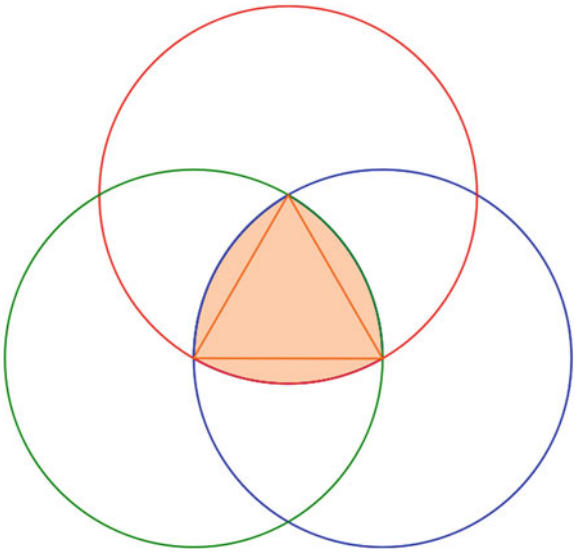


Fig. 4 A Reuleaux *triangle*, formed by the intersection of *three circular discs*. It has *central regular triangle* surrounded by *three curved regions*

regardless of the direction of those two parallel lines. For example, a circle also has constant width equal to its diameter.

It is shown in [11] that a reuleaux triangle field of width r is guaranteed to be k-covered with k sensors, where r is the sensing range of the sensors (which is same as the width r if reuleaux triangle). Intuitively, one would try to extend this result for 3D field. However, as opposed to reuleaux triangle which allows for perfect tiling in 2D space, its 3D counterpart—reuleaux tetrahedron (Refer to Fig. 6. for illustration) neither tiles in 3D space nor has a constant breadth r. Conway and Torquato in [23] gave two arrangements of regular tetrahedra such that five regular tetrahedral packed around a common edge would yield a gap of 7.36°, and twenty regular tetrahedra packed around a common vertex yield gap of 1.54 steradians. Thus the analysis of k-coverage in 2D space is not easily extendable to 3D space.

By dividing each of the halves of sensing spheres into six congruent 3D regions called slices, with each slice having three flat faces and one curved face representing an equilateral spherical triangle, Ammari et al. [10] shows that if a sensor is located in region <A,C,D> it is able to cover the whole slice as shown in Fig. 5. Any other location of active sensors would result in coverage-less patches. Thus to efficiently solve k-coverage problem with minimum number of spheres, all the active spheres have to be located in region <A,C,D>.

The basic foundation concept in [9] uses the fact that maximum overlap volume of four sensing spheres such that every point in this overlap volume is 4-covered requires that each sensing sphere is at distance r from the centers of all other three sensing spheres. The intersection volume forms reuleaux tetrahedron denoted by RT(r). Reuleaux tetrahedron does not have constant width and maximum distance between pair of points on boundary of RT(r) is 1.066r. Refer to [68] for rigorous

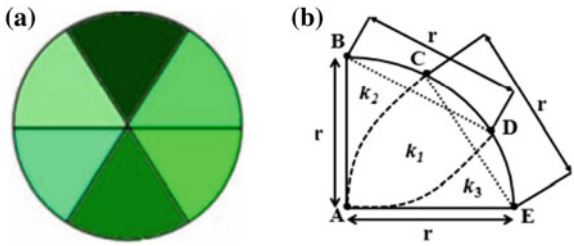


Fig. 5 **a** 2D projection of *half sphere* and its *six slices*. **b** 2D projection of a slice

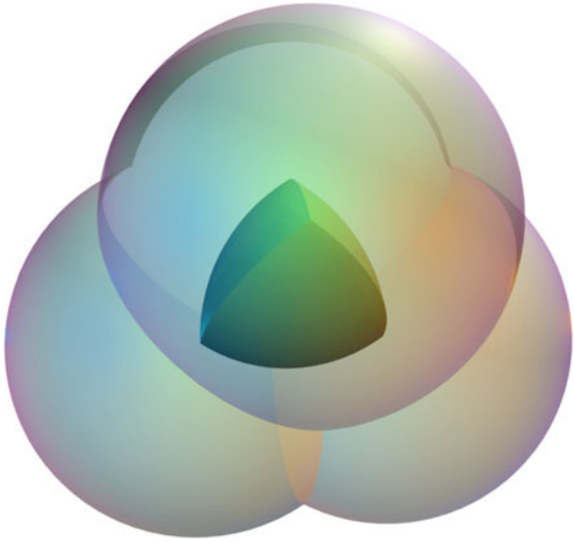


Fig. 6 Reuleux tetrahedron formed by intersection of *four symmetric spheres*

proof. For k-coverage the maximal distance between all the pair of points should not exceed r—the radius of sensing spheres. Therefore for k-coverage RT(r) should have side length $\mathrm{r}_0 = r/1.066$. The results for minimal spatial density for k-coverage and corresponding connectivity k(G) are listed in Table 4.

Table 4 Relationships for spatial density and connectivity for k- coverage when realeaux tetrahedron-based deployment

Minimal spatial density required to fully k-cover 3D field	$\lambda(r,k) = \frac{k}{0.422r_0^3}$
For homogeneous 3D k-covered field, the connectivity k(G) with G being communication graph, $a = \mathrm{R}/r$	$12.024a^3k \leq k(\mathrm{G}) \leq \frac{\mathrm{R}\mathrm{V}^{2/3}k}{0.422r_0^3}$

3.2.2 Spherical Cap-Based Coverage Analysis

Since sensing range of a node is visualized by sphere, it can be said that as long as spheres of all sensors are sufficiently covered, the whole monitored 3D field is sufficiently covered. The problem then breaks down to checking whether each sphere is sufficiently covered. Huang et al. [21] adopts the strategy of reducing the geometric problem from 3D to 2D space and then further to 1D space. This results in simplified model and less computationally intensive solution. It is achieved by analyzing the spherical caps formed by intersection of the spheres, and then projecting it in one dimension.

Since sensing field is divided into a number of subspaces by sensing spheres, considering the continuity nature of the 3D network, the level of coverage of a subspace can be derived from its spherical segments [21]. If each sphere is *k*-covered then the sensing field is also *k*-covered.

To elaborate on the concept consider a sensor s_i having a Ball Centered sensing zone B_i of radius r_i. The boundary of B_i gives the sphere S_i. A Ball usually refers to space inside a sphere. Open ball does not include boundary while closed ball refers to inclusion of boundaries also. In mathematical contexts where *ball* is used, a *sphere* is usually assumed to be the boundary points only (namely, a spherical surface in three-dimensional space) [67]. Refer to Fig. 7 for basic definitions.

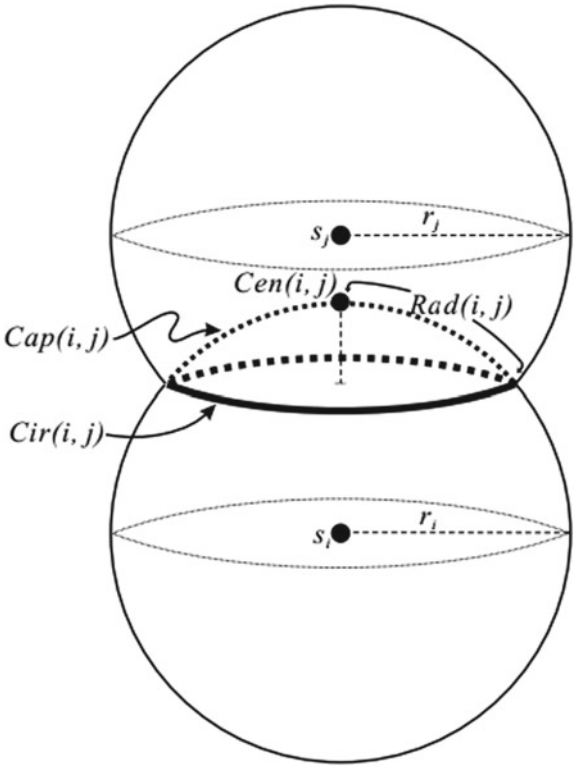


Fig. 7 The *spherical cap Cap(i,j)* is the intersection of the *sphere* S_i and ball B_j. The *circle Cir(i,j)* is the intersection of the *sphere* S_i and S_j. The center of the *spherical cap Cap(i,j),* denoted by *Cen(i,j)* is the intersection of *line* $s_i s_j$ and *Cap(i,j)*

By using simple geometric principals, it is shown in [21] that to determine the coverage in field A, it is reasonably sufficient to just find how each circle is covered. After each cap's circle coverage level is determined, the sensor's sphere coverage level can be found out, which in turn gives the overall coverage of A.

The contribution of Huang et al. [21] is to simplify a 3D problem to 2D and then further break it down in 1D resulting in reduced computational complexity and solution determination in polynomial time.

Huang et al. [37] builds on his work in [21] by elaborating on the algorithm to determine whether a sphere is *k*-covered or not. First each sensor has to check its neighboring sensor intersection with itself and calculate the corresponding spherical caps. Once the coverage degree of the each cap's circle is found, the coverage degree of the sphere can be determined, which in turn could be used to infer the coverage degree of the complete 3D field.

4 Fundamental Characteristics and Extremal Properties

For any randomly deployed 3D WSN, there are various critical transmission/sensing ranges to achieve specified degree of coverage, convergence and connectivity. In this section we briefly visit the relationship between these critical ranges and their corresponding degrees of coverage and connectivity.

Ravelomanana [51] outlays several fundamental characteristics of randomly deployed 3D WSNS for coverage and connectivity Traditionally, sensor network is modeled by visualizing n homogeneous nodes being randomly deployed in region R with Volume V = |R|, uniformly and independently [33, 52, 53]. Since number of nodes in sensor networks is usually many folds the number in ad hoc networks, and sensor network nodes are usually deployed inside a phenomenon, therefore it can be inferred that positions of nodes need not be predetermined or planned. By assuming sensing range of nodes to be spherical with radius R or R_{SENSE}, a region R (subset of IR^3) is said to be covered if every point in R is at most R_{SENSE} from a node. Two nodes can only communicate if they are within $R_{TRANS}$of each other. $\rho = n/V$ represents expected number of nodes per unit volume. By outlaying a set of theorems in [51] the following fundamental results are deduced.

Connectivity Regime and Minimum Transmission Range: If n nodes are uniformly placed in R (bounded and connected set of IR^3), then the network formed by adding edges between nodes $R_{TRANS} = \sqrt[3]{3\left(lnn + w(n)\right)/4\pi\rho}$ is connected only if $lim_{n\to\infty} w(n) \to \infty$.

Coverage and Minimum Sensing Range : If n sensors with sensing range R_{SENSE} are uniformly and independently distributed at random in R (bounded set of IR^3 of Volume V) with

$$R_{SENSE} = \frac{\sqrt[3]{3\left(lnn + lnlnn + w(n)\right)V}}{4\pi n}$$

then every point in R is covered by at least one sensor when $w(n) \to \infty$.

Table 5 Relationships between different sensing ranges and resultant coverage

Basic condition	Sensing range satisfying this relationship	Resultant condition (if and only if)
For fixed integer $l > 0$, R is l covered	$4/3\pi\rho R^3_{SENSE} = lnn + llnlnn + w(n)$	$1 \ll w(n) \ll lnlnn$
For $c(n), 1 \ll c(n) \ll \frac{lnn}{lnlnn}$	$4/3\pi\rho R^3_{SENSE} = lnn + c(n)lnlnn$	Each point covered by at least $c(n)$ spheres and at most $e\ln n$ spheres
For real number $l > 0$	$4/3\pi\rho R^3_{SENSE} = (1+l)lnn$	$\frac{-X}{W_{-1}} + o(lnn) \leq N(p) \leq \frac{-X}{W_0} + o(lnn)$; $where X = \frac{llnn}{\left(-\frac{l}{e(1+l)}\right)}$ and $N(P)$ is number of sensing nodes in each point p of R
For any $c(n), 1 \ll c(n) \ll \frac{n}{lnn}$	$4/3\pi\rho R^3_{SENSE} = c(n)\,lnn$	$N(p)$ *is covered by* $c(n)\,lnn$

Degree of Coverage: Different applications and deployment scenarios may require different degree of convergence. k degree of convergence indicates that each point p in bounded region R is covered by at least k sensing spheres of radius R_{SENSE}.

If n sensors with sensing range R_{SENSE} are uniformly and independently distributed at random in R (bounded set of IR^3 of Volume V) Then as , following holds as rigorously proved in [51]. Refer to Table 5.

$W_0(z)$ and $W_{-1}(z)$ is branch[3] of Lambert W(z) functions for W(z)≥ -1 and W(z)≤ -1 respectively.

Network degree of connectivity is also obtained simply replacing $R_{SENSE}$with R_{TRANS} in the above relationships.

Hop Diameter: The hop diameter between two nodes u and v is the shortest path needed to traverse (in terms of hops) to reach v from u or vice versa. For n sensor nodes randomly deployed in cubic region of Volume V of IR^3 with $R_{\mathbf{TRANS}} = \frac{\sqrt[3]{(3(1+l)\ln V)}}{4\pi n}$, for $l > 11/5$ then D satisfies $\boldsymbol{lim}_{\mathbf{n}\to\infty}[\mathbf{D} \leq 123\,\pi n (61 + l)\,\mathbf{ln}n] = 1$.

5 Conditional Connectivity and Forbidden Faulty Set

A node failure adversely affects the coverage and connectivity regime of any network including 3D WSNs. For 3D WSNs models to be more realistic it is essential that node failure is included in analysis. The concept of conditional connectivity and forbidden faulty sensor set studies the effect of sensor failures on 3D k-coverage of the network. Ammari et al. [9] also explores the concept of conditional connectivity [35] and forbidden faulty sensor set [26] both for homogeneous and heterogeneous k-covered

3D WSNs. The work in [9] outlays model for both conditional and unconditional connectivity.

In an attempt to better quantify the fault resilience of a network, the concept of forbidden faulty sets has been introduced by Esfahanian in [26]. The idea states that each node has at least one non-faulty neighbor. Under this forbidden faulty set condition, the number of tolerable faulty nodes is significantly larger with a slight increase in the fault diameter [24]. The forbidden faulty set analysis of restricted connectivity and fault tolerance assumes that a set of nodes cannot be faulty at the same time. For homogeneous networks having k-coverage degree in a cubic field, every location must be k-covered including the location ξ_0 of the sink s_0. To disconnect sink under the forbidden faulty set constraint, the reuleaux tetrahedron (with side r_0) should be surrounded by an empty annulus (sensors in this annulus have gone faulty) of width R. It can be visualized that reuleaux tetrahedron and annulus together they form a larger (outer) reuleaux tetrahedron of side $r_0 + 2R$. The conditional connectivity for such a network then becomes

$$k\,(G) = \frac{((r_0 + 2R)^3 - r_0^3)k}{r_0^3}$$

It is important to understand that any non-faulty sensor located in the inner RT still has non-faulty neighbors in the inner reuleaux tetrahedron. The same holds for any non-faulty sensor located outside the outer reuleaux tetrahedron i.e. it has non-faulty neighbors outside outer reuleaux tetrahedron. Any faulty sensor inside the annulus has non-faulty neighbor either in inner or outer reuleaux tetrahedron.

For heterogeneous 3D k-covered networks the annulus may contain sensors with communication radii less than $R_{max}/2$, the non-faulty neighbor sensors in one connected component will only be able to communicate with non-faulty neighbors of other connected component if width of annulus is less than R_{max} [9].

6 *k*-Connectivity and Full Coverage

k-connectivity and full coverage has many important real world applications in the domain of 3D Wireless Sensor Networks. k-connectivity indicates that there are at least k disjoint paths between any pair of sensor nodes. Though intuitively a network designer may want maximal connectivity for a network which may result in increased resilience in case of node failure, however, maximal connectivity comes at the cost of increased number of deployed nodes or increased spatial density of deployed sensors. Therefore network designers always have to find a sensible tradeoff between k-connectivity of network and practicality. The final choice of k-connectivity for a network boils down to particular application specific requirements, deployment constraints and economics.

6.1 Low Connectivity and Full Coverage (Lattice Pattern)

Low connectivity indicates that there are at least k disjoint paths between any pair of sensor nodes where $k \leq 4$.

Many airborne sensor network applications use low connectivity deployment strategy. Since the scale and volume of deployment field is huge in most such applications, the choice of low connectivity and full coverage model makes economic and practicality sense. Some of the examples include airborne WSNs for supporting intelligent computer vision [18], helping overcome human paropsia [65], airborne defense systems [1], and atmospheric pollution monitoring [59]. The applications however are not only limited to airborne 3D WSNs but also include submerged WSNs.

Zhang et al. [74] and Bai et al. [16] study the problem of low connectivity and full-coverage in the domain of lattice. By using regular lattice deployment patterns in [74], the optimality of 1-connectivity, 2-connectivity, 3-connectivity and 4-connectivity for any value of R_{TRANS}/R_{SENSE} or r_c/r_s is derived. All sensors have same spherical communication r_c and sensing r_s domains and are deployed in vast 3D field such that boundary problems can be ignored.

6.1.1 Lattice Pattern for 1- AND 2-Connectivity

Due to symmetry of lattice odd connectivity patterns do not exist in this scenario. Therefore the optimal solution (less sensor nodes required in a pattern) for 1-connectivity is also optimal solution for 2-connectivity.

Zhang et al. [74] and Bai et al. [16] address this problem by categorizing the r_c/r_s in four different ranges. The results are summarized in Table 6.

6.1.2 Lattice Pattern for 3- AND 4-Connectivity

The lattice model for 1-and 2- connectivity is extended for 3- and 4- connectivity and full coverage in [74] and [16] (Table 7).

Detailed analysis in [74] summarizes that when *rc/rs* = 1, the number of nodes needed to achieve 14-connectivity is around 2.5 times that to achieve 3- or 4-connectivity, and 3.5 times that to achieve 1- or 2-connectivity. The extra number of nodes required for 14-connectivity increases as *rc/rs* decreases. When *rc/rs* = 0.5, the number needed to achieve 14-connectivity is almost 6 times that to achieve 3- or 4-connectivity, and around 18 times that to achieve 1- or 2-connectivity as indicated in Fig. 10.

Table 6 Lattice pattern 1- and 2- connectivity generated for different ranges of r_c/r_s

r_c/r_s Range	Lattice pattern generated	Dimensions of lattice
$r_c/r_s < 4/3$	Body centered lattice generated by cuboid a Height of a is rc Refer to $\Lambda 2$–1 in Fig. 8 for illustration	Upper and lower faces lengths $e1 = \sqrt{\frac{1}{2}\left(3r_s^2 - r_c^2 + r_s\sqrt{9r_s^2 - 2r_c^2}\right)}$
	Any sensor can communicate with neighbors along direction of height	$e2 = \frac{1}{2}\left(3r_s + \sqrt{9r_s^2 - 2r_c^2}\right)$
$4/3 \leq r_c/r_s < 12/\sqrt{9+32\sqrt{3}}$	Body centered lattice generated by cuboid a Refer to $\Lambda 2$–2 in Fig. 8. Height of a is rc. Any sensor can communicate with neighbors as shown in $\Lambda 2$–2	Upper and lower faces lengths $e3 = e4 = \left(\sqrt{4r_s^2 - r_c^2/4}\right)$
$12/\sqrt{9+32\sqrt{3}} \leq \frac{rc}{rs} < 2\sqrt{3}/\sqrt{5}$	Body centered lattice generated by cube a Refer to $\Lambda 2$–3 in Fig. 8. Any sensor is able to connect with its two neighbors along the direction of B-diagonal.	Length of edges $e5 = 2r_c/\sqrt{3}$
$2\sqrt{3}/\sqrt{5} \leq \frac{rc}{rs}$	Body centered lattice generated by cube a Refer to $\Lambda 2$–4 in Fig. 8. Any sensor is able to connect with its two neighbors along the direction of B-diagonal	Length of edges $e6 = 4r_s/\sqrt{5}$

6.2 High Connectivity and Full Coverage (Lattice Patterns)

6.2.1 Lattice Pattern for 6- AND 14-Connectivity

In work similar to [16, 74], Bai et al. [15] outlay a set of patterns for full coverage and two representative connectivity requirements, i.e. k = 14-, 6-connectivity.

For 14-connectivity, the lattice pattern follows body centered cubic (bcc) lattice generated by cube ABCDEFGH with center O. Without the loss of generality, the sensor position at O has 8 of its connected neighbors as vertices of the cube ABCDEFGH and other 6 are the centers of neighboring cubes IJKLMN. The voronoi polyhedron generated by each sensing sphere in such pattern is truncated octahedron.

For further details refer to Table 8 and Figs. 11 and 12 summarizing the approach in [15].

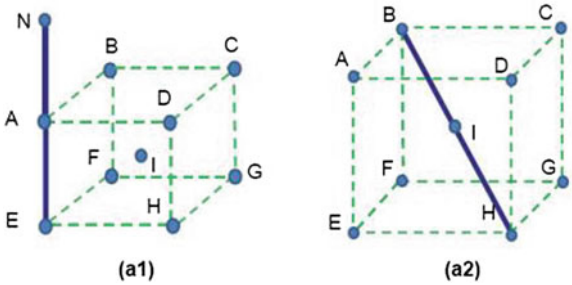


Fig. 8 Lattice patterns for 1- and 2-connectivity with full coverage. For $\Lambda_{2\text{-}1}(r_c/r_s < 4/3)$and $\Lambda_{2-2}(4/3 \leq r_c/r_s < 12/\sqrt{9+32\sqrt{3}})$ refer to (**a1**). $\Lambda_{2\text{-}1}$and Λ_{2-2}have the same lattice structure (*cuboid*) but have different edge lengths which are dictated by the ratio of r_c/r_s. The communication in this body centered lattice structure is along the height of the *cuboid*. For $\Lambda_{2-3}(12/\sqrt{9+32\sqrt{3}} \leq \frac{rc}{rs} < 2\sqrt{3}/\sqrt{5})$(and $\Lambda_{2\text{-}4}(2\sqrt{3}/\sqrt{5} \leq \frac{rc}{rs})$ refer to (**a2**). $\Lambda_{2\text{-}3}$and $\Lambda_{2\text{-}4}$ have same lattice structure (*cube*) but different edge lengths which are dictated by the ratio of r_c/r_s. The communication in a2 is along the B-Diagonal of lattice structure. The illustration has been taken from [74]. Refer to [74] for further details

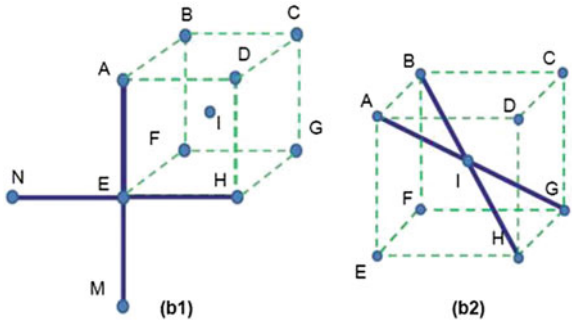


Fig. 9 Lattice patterns for 3- and 4-connectivity with full coverage. For $\Lambda_{4\text{-}1}(r_c/r_s < 4/3)$ and $\Lambda_{4\text{-}2}(4/3 \leq r_c/r_s < 2\sqrt[3]{2}/\sqrt{3})$ refer to (**b1**).$\Lambda_{4\text{-}1}$and $\Lambda_{4\text{-}2}$have the same lattice structure (*cuboid*) but have different edge lengths which are dictated by the ratio of r_c/r_s. The communication in this body centered lattice structure is along the height plane of the *cuboid*. For $\Lambda_{4\text{-}3}(2\sqrt[3]{2}/\sqrt{3} \leq \frac{rc}{rs} < 2\sqrt{3}/\sqrt{5})$ and $\Lambda_{4-4}(2\sqrt{3}/\sqrt{5} \leq \frac{rc}{rs}))$ refer to (**b2**). $\Lambda_{4\text{-}3}$and Λ_{4-4} have same lattice structure (*cube*) but different edge lengths which are dictated by the ratio of r_c/r_s. The communication in b2 is along the B-Diagonal Plane of lattice structure. The illustration has been taken from [74]. Refer to [74] for further details

6.3 Lattice Pattern Interdependency and Mutual Relationships (Pattern Evolution)

It is interesting to observe that how different patterns are linked. By the principal of symmetry lattice Λ2-2 also yields 4-connectivity and Λ2-3 yields 8-connectivity (sensor I can connect with sensors A, B, C, D, E, F, G, and H). Similarly Λ2-4 also achieves 14-connectivity. Λ4-3 achieves 8-connectivity and Λ4–4 achieves 14-connectivity. Furthermore it can be analyzed that if a certain lattice pattern is optimal

Table 7 Lattice pattern for 3- and 4-connectivity generated for different ranges of r_c/r_s

r_c/r_sRange	Lattice pattern generated	Dimensions of lattice
$r_c/r_s < 4/3$	Body centered lattice generated by cuboid a Height of a is rc Refer to $\Lambda 4$–1 in Fig. 9 for illustration Any sensor can communicate with its 4 neighbors in one plane. sensor E is connected with A, N, M and H	Upper and lower faces lengths $e7 = e8 = \left(2r_s + \sqrt{4r_s^2 - 2r_c^2}\right)$
$4/3 \leq r_c/r_s <$	Body centered lattice generated by cuboid a Refer to $\Lambda 4$–2 in Fig. 9. Height of a is $4r_s/3$ sensor E is connected with A, N, M and H	Upper and lower faces lengths $e9 = 4r_s/3\ e10 = 8r_s/3$
$2\sqrt[3]{2}/\sqrt{3} \leq \frac{rc}{rs} < 2\sqrt{3}/\sqrt{5}$	Body centered lattice generated by cube a Refer to $\Lambda 4$–3 in Fig. 9. sensor I is connected with A, B, G and H	Length of edges $e5 = 2r_c/\sqrt{3}$
$2\sqrt{3}/\sqrt{5} \leq \frac{rc}{rs}$	Body centered lattice generated by cube a Refer to Λ 4–4 in Fig. 9. Any sensor is able to connect with its two neighbors along the direction of B-diagonals.sensor I is connected with A, B, G and H	Length of edges $e6 = 4r_s/\sqrt{5}$

solution for *k1* connectivity and also yields *k2* connectivity where $k2 > k1$, then it is also optimal solution for *k2* connectivity. Therefore Λ2-2 , Λ2-3 , Λ4-3 , Λ2-4 and $\Lambda 4 - 4$ are optimal lattice patterns for up to 4, 8, 8, 14 and 14 respectively [16, 74]. Similarly Λ6-2 pattern [15] actually achieves $8-$connectivity, and Λ6-3 pattern achieves 14-connectivity for $4\sqrt{5} \leq rc/rs$.

6.4 Full Coverage and 1-Connectivity (Strip-Based Approach)

A strip-based placement scheme is proposed for 3D networks in [8] that provides full coverage and 1-connectivity when $\frac{rc}{rs} < 4/\sqrt{5}$. The approach will automatically provide full coverage and *k*-connectivity for $\frac{rc}{rs} > 4/\sqrt{5}$ Based on work in [14] for 2D networks, Alam et al. [8] extends the concept for 3D networks.

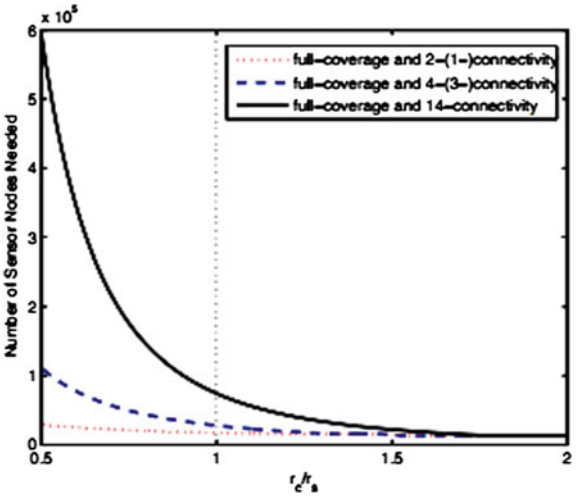


Fig. 10 Number of sensors nodes for 2- (1-), 4- (3-) and 14-connectivity by optimal lattice patterns, respectively, plotted against different ratios of r_c/r_s. Deployment volume: $1000^3\,m^3$.r_s = 30 m.15 m < r_c < 30 m. [16]

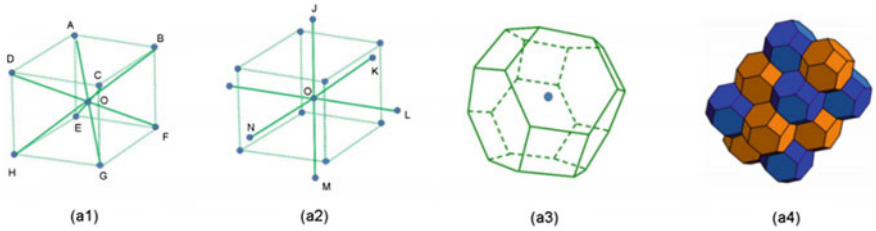


Fig. 11 *Solid lines* represent connected links and *dashed lines* represent body structure of cube. For 14- connectivity for the sensor located at O, **a1** shows 8 connected neighbors which are vertices of the cube, and **a2** shows other 6 connected neighbors which happen to be the center points of the neighboring cubes. **a3** shows the Voronoi polyhedron formed by the sensing spheres of the connected sensor nodes: truncated octahedron. **a4** The lattice pattern achieves full coverage of a 3D volume, which is illustrated by Voronoi polyhedra generated by sensing spheres. Illustrations taken from [15]

By setting distance between any two nodes in a strip as $a = min\{r_c, 4r_s/\sqrt{5}\}$ and distance between two parallel strips in a plane as $\beta = 2\sqrt{r_s^2 - (a/4)^2}$; and distance between two planes of strips as $\beta/2$, and deploy strips such that strip of one plane is placed between two strips of neighboring planes as shown in Fig. 13.

The distance between two nodes residing in different planes then becomes γ, Where $\gamma = \sqrt{\beta^2/2 + a^2/4}$. Since maximum distance between any two nodes in

Table 8 Lattice pattern for 6- and 14-Connectivity generated for different ranges of r_c/r_s

r_c/r_s Range	Lattice pattern generated	Dimensions of lattice
For $k = 14$	Body centered cubic lattice generated, center at O. 8 neighbors are vertices, and remaining 6 are centers of adjacent cubes. Refer Fig. 11	Length of edge $e5 = min(4r_s/\sqrt{5}, r_c)$ voronoi polyhedron: truncated octahedron
For $k = 6 r_c/r_s < 9\sqrt{43}$	Pattern is basic lattice, denoted by Λ6-1. Generated by seed parallelepiped a with diamond shaped-base. Its six neighbors are ABCDE and F. Refer Fig. 12	$e_1 = r_c$ one base diagonal = rc $h = 2\sqrt{(2r_c r_s^2 + r_s^3 - r_c^3) / (2r_c + r_s)}$ voronoi polyhedron: hexagonal prism
$9/\sqrt{43} \leq r_c/r_s < 2\sqrt{3}/\sqrt{5}$ For $k = 6$	Pattern is bcc, denoted Λ6-2. Refer Fig. 12	*Edge length* $e2 = 2r_c/\sqrt{3}$
$2\sqrt{3}/\sqrt{5} < rc/rs$For $k = 6$	Pattern is bcc, denoted Λ6-3. Refer Fig. 12	$e3 = 4r_s/\sqrt{5}$

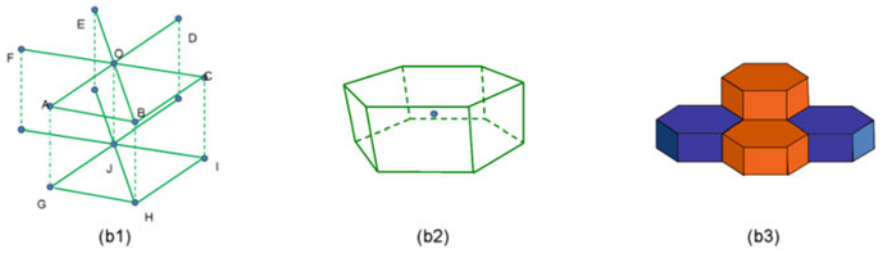


Fig. 12 Lattice patterns for 6-connectivity with full coverage. When $r_c/r_s <$, the pattern formed is shown in **b1**, the sensing *sphere* form voronoi polyhedron called hexagonal*prism***b2**, **b3** shows that full coverage is achieved in 3D environment. For $r_c/r_s <$ and , the pattern formed is body centered *cube* similar to (**a1**) in previous figure with edges and respectively

original truncated octahedron (TO) model is 4 rs /$\sqrt{5}$, connectivity is maintained when by setting a = min{r_c, $4r_s/\sqrt{5}$} Since only 1-connectivity is required, β can be kept as large as possible as long as full coverage is not compromised.

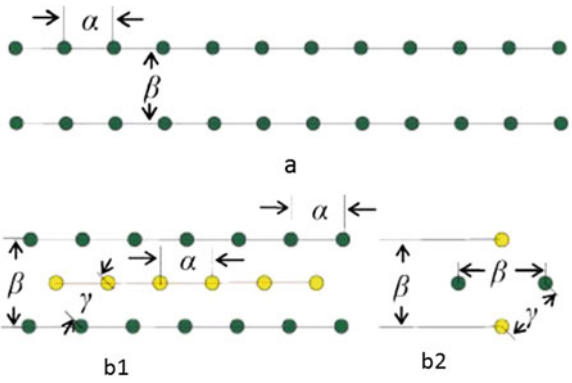


Fig. 13 **a** Nodes in a particular plane. **b** horizontal and **c** vertical projections nodes in two different *planes*. Nodes with same color are from same *planes* [8]

7 Continuum Percolation

Let P be a homogeneous Poisson Process in k-dimensional Euclidean Space, R^k. Let S be a random *k*-dimensional shape, often a sphere. Centre an independent copy of S at each point of P. Percolation occurs with positive probability if any given random shape is part of infinite clump of random shapes [34]. In the theory of continuum percolation, nodes are distributed according to Poisson density λ. The theory states that there exists a positive value of λ commonly known as critical density, λ_c, for which phase transition occurs in graph. In simplicity, it indicates that node density of λ_c is required to achieve network detectability close to 1—i.e. when an event takes place within a sensor network, its detection probability is almost P. Hall [34] comprehensively addresses the problem of continuum percolation. Gilbert [30] introduced continuum percolation as a model for growth and structure of random networks in communication theory. The target is to find the critical density of a Poisson point process at which an unbounded connected component almost surely appears so that the network can provide long distance multi-hop communication.

Ammari and Das [12] investigate the problem of the critical density for percolation in coverage and connectivity in 3D WSNs, as well as the corresponding critical network degree. Following three questions are addressed:

(a) Critical density above which a giant covered region of a field will almost surely appear for the first time and its corresponding critical network degree.
(b) Critical density above which a giant connected component will almost surely appear for the first time and its corresponding critical network degree.
(c) Critical density above which a giant covered region of a field and giant connected component will almost surely appear for the first time and its corresponding critical network degree.

The work in [12] explains that due to dependency between coverage and connectivity in WSNs, the problem is not only a continuum percolation but also integrated continuum percolation. Since generally speaking communication and sensing radii of the nodes are not same, hence critical density for full coverage does not necessarily

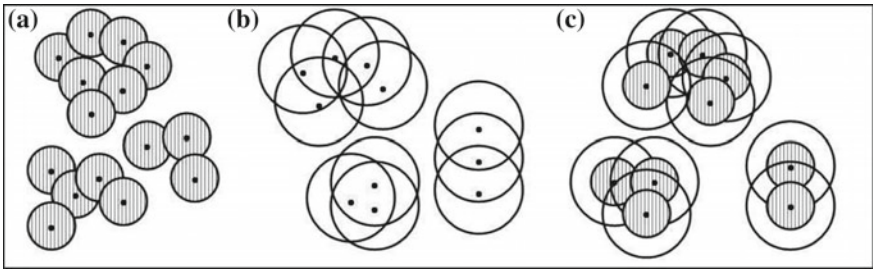


Fig. 14 **a** covered, **b** connected, **c** coordinated components [12]

mean it will result in complete connectivity of the network. Following definitions excerpted from [12] will help understand further text.

A covered component (or covered region) is a maximal set of sensing spheres (i.e., not included in any other subset except when it is equal to the original entire set of sensing spheres) whose corresponding sensors are collaborating directly or indirectly (Fig. 14a shows three covered components).

A connected component is a maximal set of communication spheres whose corresponding sensors are communicating directly or indirectly (Fig. 14b shows three connected components).

A coordinated component is a maximal set of concentric sensing and communication spheres whose corresponding sensors are coordinating directly or indirectly (Fig. 14c shows three coordinated components).

Let $X_\lambda = \{\xi_i : i \geq 1\}$ be a homogeneous Poisson Point of density λ in R^3, where ξ_i is the location of sensor s_i. The sensing range of s_i is a sphere of radius r_i and communication range is a sphere of radius R_i, centered at ξ_i (location of s_i). All deployed sensors have same communication and sensing radii (homogeneous model).

$$B_i(r) = \{\xi \Lambda R^3 : |\xi_i - \xi| \leq r\}$$
$$B_i(R) = \{\xi \Lambda R^3 : |\xi_i - \xi| \leq R\}$$

where $|\xi_i - \xi|$ is the Euclidean distance between ξ_i and ξ.

Two spheres s_i and s_j are said to be collaborating if $|\xi_i - \xi_j| \leq 2r$; and communicating if $|\xi_i - \xi_j| \leq R$. They are said to be coordinating if only if they both collaborate and communicate.

Since percolation model can be visualized as an ensemble of points distributed in space, where some points are adjacent or connected, Boolean Model is considered:

A Boolean Model consists of two components, namely, point process X_λ and connection function: h. The set $X_\lambda = \{\xi_i : i \geq 1\}$ is a homogeneous Poisson point process of density λ in 3D Euclidean space R^3 where elements of X_λ are the location points of sensors. The connection function h states that two points ξ_i, ξ_j are adjacent with probability $h|\xi_i - \xi_j| = 1 \text{if} |\xi_i - \xi_j| \leq d$, and $h|\xi i - \xi j| = 0 \text{if} |\xi_i - \xi_j| > d$ where $d \geq 0$ and $|\xi_i - \xi_j|$ is the euclidean distance between ξ_i and ξ_j [12].

7.1 Sensing Coverage Percolation COVP

For a given 3D field, compute the density λ_c^{cov} called the critical coverage percolation density such that there surely exists a giant covered region that spans the entire sensor field when $\lambda > \lambda_c^{cov}$. It is noteworthy that Boolean function $(X_\lambda, \{B_i(r) : i \geq 1\})$ only percolates when $\lambda > \lambda_c^{cov}$. Refer to [12] for rigorous proof. We summarize the results in the Table 8.

7.2 Network Connectivity Percolation CONP

For a network which is originally disconnected, compute the density λ_c^{con} called the critical connectivity percolation density such that there surely exists a giant connected component that spans the entire sensor field when $\lambda > \lambda_c^{con}$. It is noteworthy that Boolean function $(X_\lambda, \{B_i(R) : i \geq 1\})$ only percolates when $\lambda > \lambda_c^{con}$. Refer to [12] for rigorous proof. We summarize the results in the Table 9.

The main difference between solutions for COVP and CONP is the compulsion for existence of sphere overlap for CONP. While for COVP it is sufficient for sensing spheres to just intersect, for CONP at least half of their communication spheres must overlap as indicated in Fig. 15.

$$V_{\min} = \frac{5}{12}\pi R^3$$

It is shown that minimum overlap volume fraction must be 0.3125. Hence $0.3125 \leq w_t < 1$ [12]; where w_t is the fraction of volume overlap of communication spheres. Similarly w_s is the fraction of volume overlap of sensing spheres. Since ideally sensing spheres just need to intersect (barely touch at boundaries), hence w_s can take range $0 < w_s < 1$.

7.3 Integrated Continuum Percolation (Coverage and Connectivity Percolation)

For a given 3D field, compute the density $\lambda_c^{cov-con}$ called the critical integrated percolation density such that there surely exists a giant covered region that spans the entire sensor field and giant connected component that spans the entire sensor field, when $\lambda > \lambda_c^{cov-con}$. Refer to [12] for rigorous proof. We summarize the results in the Table 9.

Since for two spheres s_i and s_j to collaborate, $|\xi_i - \xi_j| \leq 2r$; and to communicate $|\xi_i - \xi_j| \leq R$. They are said to be coordinating if $|\xi_i - \xi_j| \leq \min\{2r,R\}$. As can be inferred the overlap volume of the sensing spheres depend on the ratio r/R. It

Table 9 Mathematical expressions for COVP, CONP and ICP

COVP	$\lambda_c^{cov}(r, w_s) = \frac{0.119}{(1-w_s)r^3}\ 0 < W_s < 1$
CONP	$\lambda_c^{con}(R, w_t) = \frac{0.955}{(1-w_t)R^3}\ 03125 < W_t < 1$
ICP	$\lambda_c^{cov-con}(r, w_s, a) = \frac{0.955}{(1-w_s)a^3r^3}$; W_s is the overlap volume between sensing spheres, $\theta(a) \leq W_s < 1$; $\theta(a) = (4+a)(2-a)/16$; $1 \leq a < 2$; r is the radius of sensing spheres

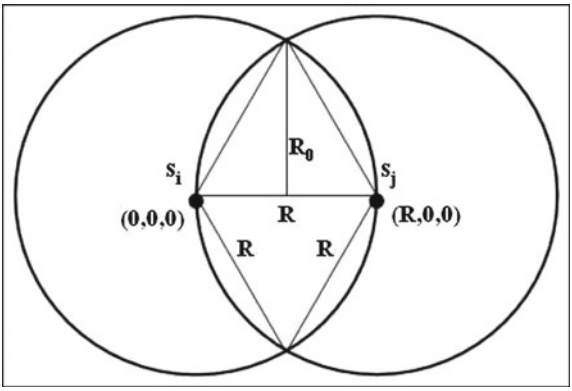


Fig. 15 Minimum overlap volume for CONP $V_{min} = \frac{5}{12}\pi R^3$

is reasonable to believe that R ≥ r since communication radius is larger than the sensing radius. Therefore R = *a*r, where a ≥ 1. Since distance between pair of any sensors cannot exceed 2r, by applying same reasoning to case when a ≥ 2, the critical percolation density for ICP is achieved. Hence, range of $a : 1 \leq a < 2$.

8 Underwater 3D WSNs: Network Planning and Deployment

Underwater sensor networks have increasingly important applications in many fields including oceanographic data collection, ocean sampling, environmental and pollution monitoring, offshore exploration, disaster prevention, tsunami and seaquake warning, assisted navigation, distributed tactical surveillance, search and rescue, and mine reconnaissance. There has been significant increase in interest to monitor aquatic environments both for scientific, environmental and defense needs [48]. Most current methods including remote telemetry and sequential local sensing cannot adequately meet the standards of submerged 3D WSNs, which require the sensing mechanism to be highly precise, real time and fine grained spatio-temporal sampling of the oceanic environment. Wireless Underwater acoustic networking is the enabling technology for these applications. Under Water Acoustic Sensor Networks (UW-ASNs) consist of a variable number of sensors and vehicles that are deployed to perform collaborative monitoring tasks over a given area. To achieve this objective, sensors and vehicles self-organize in an autonomous network which can adapt to the characteristics of the ocean environment [60].

Acoustic communications is the typical physical layer technology for most underwater sensor networks. The choice is justified by the fact that radio waves propagate at long distances through conductive sea water only at extra low frequencies (30–300 Hz). This requires impractically large antennas and high transmission power—not a friendly design choice for a wireless sensor node. For example, the experiments performed by Robotic Embedded System Laboratory (RESL) at University of Southern California showed that Berkeley Mica 2 Motes—a popular WSN node platform—is reported to have transmission range of only 120 cm underwater at 433 MHz [60]. Optical signals although do not suffer from such high attenuation but are affected by scattering, and also require high precision to establish laser beam communication link between two small nodes. Therefore acoustic wireless communication is the only reasonable choice for underwater sensor networks. However, design of an underwater sensor network is prohibitively demanding task both in terms of technological and practical constraints. UW-ASN poses unique challenges due to harsh underwater environment, such as limited bandwidth capacity [54], high and variable propagation delays [50], high bit error rates, and temporary losses of connectivity caused by multipath and fading phenomena [56].

There are two fundamental deployment architectures for UW-ASNs, i.e., two-dimensional architecture, where sensors are anchored to the bottom of the ocean, and the three-dimensional architecture, where sensors float at different ocean depths covering the entire monitored volume region. A three-dimensional deployment strategy is usually adopted when the phenomena cannot be adequately observed by bottom-anchored sensor network [3]. Younis et al. [73] gives a detailed guide on node placement in wireless sensor networks. Though the work focuses on 2D WSNs, it outlays some essential considerations for node placement which are also relevant to 3D Network Architectures.

UW-ASN network designer has following objectives to meet:

(1) Determine the minimum number of sensors to be deployed while not compromising the target coverage (sensing) and connectivity.
(2) Provide guidelines on how to choose the optimal deployment surface area, given a target region.
(3) Study the robustness of sensor network to node failures, and include appropriate number of redundant sensors in the design to account for node failures.

In [51], connectivity and coverage in 3 dimensional networks is extensively studied. It is shown that sensing range r required to achieve *1*-coverage is greater than transmission range t required to achieve connectivity. Since in most applications $t > r$, hence a network is guaranteed to be connected if it has at least *1*-coverage.

Pompili et al. [48] and [49] extensively analyze the deployment strategies for underwater acoustics networks. While the major focus of the work is in 2D domain, [48] also addresses the major problems for 3D UW-ASNs. In 3D WSN sensors float at different depth to observe a phenomenon. Usually sensors are deployed by anchoring winch-based sensor devices to the bottom of the ocean as depicted in Fig. 16b. Each sensor is equipped with a floating buoy which pulls the sensor towards ocean surface. The length of wire connecting the sensor to the anchor is used to control the depth

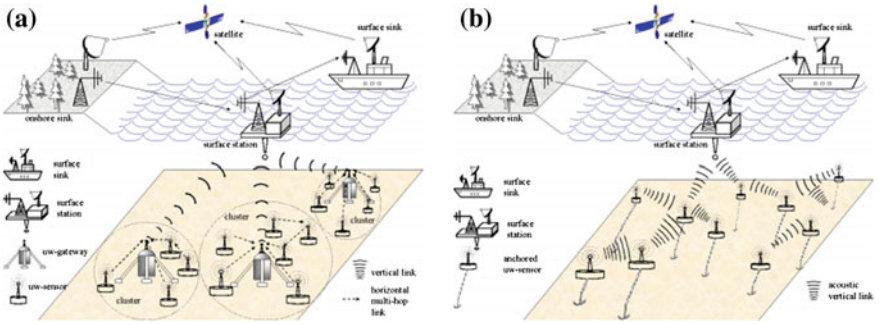


Fig. 16 Architectures for **a** two-dimensional and **b** three-dimensional underwater sensor networks. Illustration is taken from [48]

of the sensor. The network sensors should regulate their depths collaboratively to enable coordinated operations, coverage of 3D field and ensure network topology be always connected.

Pompili et al. [48] proposes three deployment strategies for three-dimensional UW-ASNs to obtain 1-coverage while enlisting following assumptions. These assumptions are relevant to most 3D UW-ASNs network designers.

(1) *Transmission range of the sensor is greater than sensing range of the sensor.*
(2) *Solution for 1-Coverage UW-ASN will also yield a connected topology of UW-ASN because transmission range of a sensor is greater than sensing range of the sensor.*
(3) *Winch-based sensor devices are anchored to the bottom of the ocean in such a way that they cannot drift with ocean currents.*
(4) *Sensor devices are equipped with floating buoy that can be inflated by a pump. Sensors therefore can adjust their heights according to requirements.*

The three deployment strategies in [48] are given below in Table 10:

For a given target coverage ratio, it can be generalized that minimum number of sensors needed to achieve desired coverage ratio decreases with the complexity of the deployment strategy.

8.1 Grid-Based Deployment Scheme

Cayirci et al. [19] and Tezcan et al. [57] use a 3D grid-based coordinate system in which the whole sensing volume is divided into cubes of size r. r is the resolution distance which is the edge length of a unit cube that sensor space is partitioned into. Coordination distance a is also specified which indicates the distance in neighboring cubes for a node to coordinate its depth. While a node arranges its depth, it exchanges information with the nodes within the coordination distance. Figure 17 illustrates the concept.

Table 10 Three major deployment strategies

3D Random	Simplest Strategy, No coordination from surface station. Sensors are randomly deployed, sensors choose their depth randomly. Each sensor informs its position	Will require greatest number of sensor nodes to cover a region. Might result in coverage-less patches. Only suitable for less critical applications
Bottom-Random	Sensors are randomly deployed at the bottom. Each sensor informs its location. Surface station then calculates the optimal depth for each sensor. Sensors then take that depth	Relatively more efficient strategy. Better 1-coverage probability is ensured
Bottom-Grid	Assisted by AUVs. Sensors deployed to pre-defined target locations to obtain grid deployment at the bottom of ocean. Each sensor then accordingly floats to its designated depth	The benefit of reduced number of sensors might be offset by expensive deployment strategy. Yields best result for 1-coverage

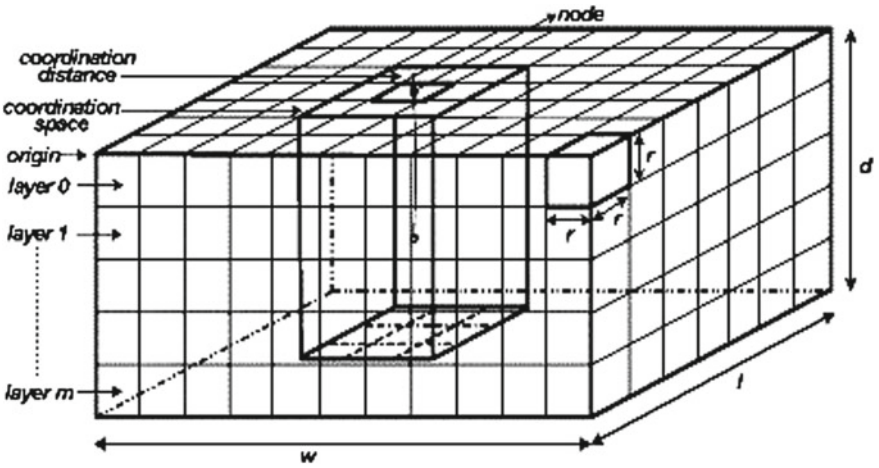


Fig. 17 The coordinate system for a sensor space with coordination distance $a = 1$ [19]

Each cube is also identified along the z-axis (depth) from the sea surface to maximal depth with a layer number. For example, all the cubes between depth 0 and depth r constitute layer 0.

The goals:

(1) Achieve Maximum Coverage Efficiency
$\varepsilon = \frac{n_c}{n_n}$; where n_c is the number of cubes covered by at least one sensor and n_n is the total number of nodes in the sensor space.
(2) Achieve Maximum Average Distance

$\theta = \frac{\sum_{i=1}^{k} e_i}{k}$; where e_i is the between two nodes in node pair i. If there are n_N nodes in the sensor field, we can have as many as k node pairs; $k = \frac{n_N(n_N-1)}{2}$. The goal is to reduce the probability that all nodes are concentrated on certain depths.

In the proposed distributed 3D space coverage scheme [19], a node coordinates its depth with the other nodes in their coordination space whose x and y coordinates meet the following conditions.

$$x_i - a \leq x_n \leq x_i + a$$
$$y_i - a \leq y_n \leq y_i + a$$

where x_n and y_n are coordinates of the given node, and x_i and y_i are coordinates of the neighboring nodes. The proposed algorithm elaborates that nodes maintain neighbor tables. Nodes listen to the neighbor broadcasts of info packets within their coordination space. Based on the available information, a node selects an appropriate depth for itself.

8.2 Cluster-Based Deployment Scheme

Akkaya et al. [2] gives a scheme for self-deployment of sensor networks for maximized coverage in underwater acoustic sensor networks. The scheme which is designed to be fully distributed has four phases: (1) Clustering, (2) Grouping, (3) Depth Assignment and (4) Additional Rounds. In Clustering, nodes, which are randomly deployed at the bottom of the ocean, are clustered with the node with highest ID becoming the cluster leader. In Grouping, Cluster leaders determine the possible coverage overlaps based on the information provided by the nodes in the cluster. The cluster leader makes sure that no two nodes having an edge shares the same GID. A sample grouping is shown in Fig. 18. Once Cluster leader finishes the process, it sends a message to each node within the cluster which contains the node's GID. The GIDs are than used to determine the new location (i.e., depth) of the nodes.

Depths are assigned in phase three with the objective of reducing the coverage overlaps and improve the overall 3D coverage. The space between two different groups will be D/(G + 1) where D is the depth of water and G is the number of groups. Finally in phase four, each node determines its closest neighbor in terms of distance and checks whether it has a sensing coverage overlap with that node. The node will move by an amount dependent on the distance to this neighbor. The closer the two nodes are, the further apart they will attempt to move from one another. The movement will need to be stopped when there is no significant improvement for coverage or a certain number of rounds is achieved.

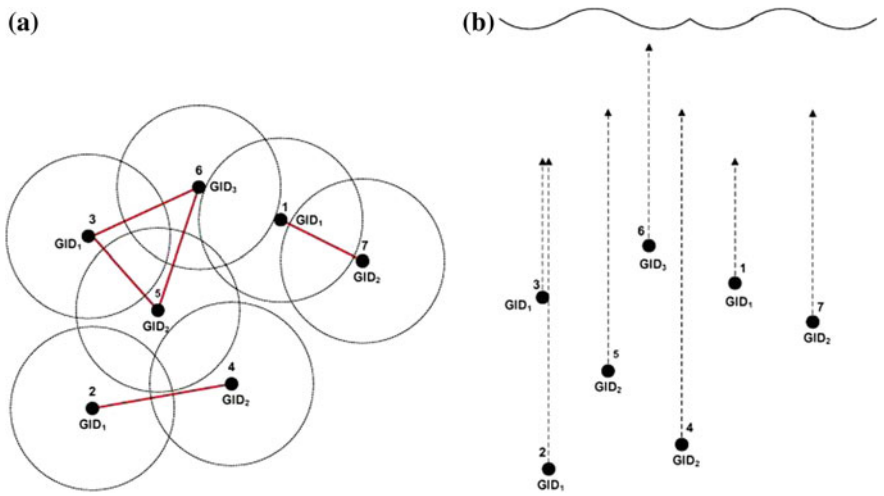


Fig. 18 **a** Grouping of the nodes based on sensing coverage overlaps. **b** Depth adjustment based on grouping [2]

8.3 Tetrahedrons-Based Deployment Scheme

In many practical underwater applications full coverage might not be possible because of vast dimensions of monitored region, then using truncated octahedrons might not be the best solution. Akkaya et al. [2] highlights interesting result which may have practical ramifications for many UW-ASNs deployments. Although it has been shown in [4] that optimal space filling polyhedron-based deployment strategy is truncated octahedron, Akkaya et al. observe that it is only true when there are enough sensor nodes to cover the complete 3D field. In such a scenario minimizing the overlap between sensing ranges will improve the overall coverage. Refer Fig. 19. For such a placement, the use of tetrahedrons would yield better results. The nodes are located at the vertices of these tetrahedrons.

8.4 Hierarchical Underwater Networks and Deployment Analysis

A hierarchical network has more powerful and robust backbone nodes and less powerful failure prone sensor nodes. A non-hierarchical network is like an ordinary sensor network where there is only one type of nodes. Alam et al. [5] analyzes node placement for hierarchical and non-hierarchical underwater networks. Only the hierarchical network work in [5] is added in this section since the concepts for non-hierarchical network are adequately covered in the earlier sections.

The network backbone nodes communicate with the sink using other backbone nodes as routers. Backbone nodes placement strategies are based on Voronoi cells

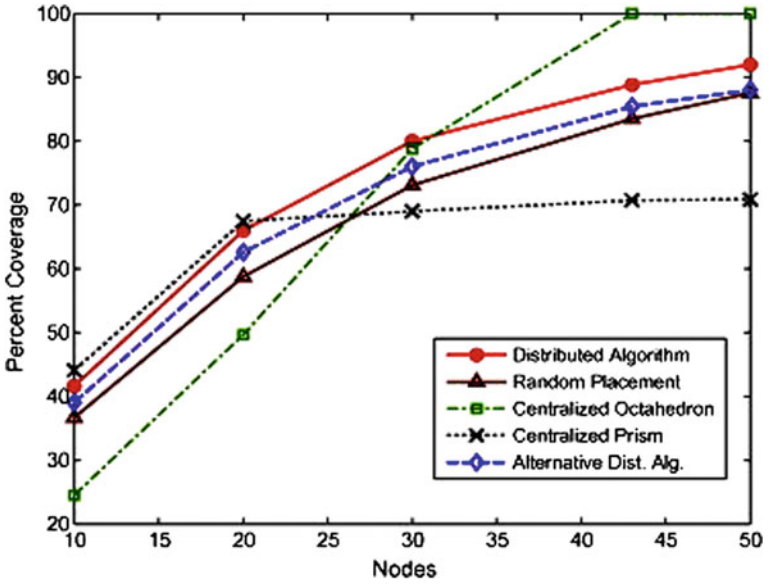


Fig. 19 Coverage comparison with varying the number of nodes. The simulation experiment used different deployment schemes to cover a fixed volume. Octahedron-based placement performs poorly for reduced number of node but yields maximum coverage when number of nodes are increased [2]

formation in 3D space. Neglecting boundary effects, ideally the total number of nodes required to fully cover the region is ratio of volume of total monitored region to volume of one Voronoi cell. Revisiting the concepts explained in earlier sections, minimizing the number of nodes can be achieved if the corresponding virtual Voronoi cell has the highest volume among all placement strategies subject to the constraint that the radius of its circumsphere cannot exceed r_{bs}, where r_{bs} is some deterministic threshold dictating the communication range between sensor node and back bone node. The analysis of different virtual Voronoi cells (cube (CB), hexagonal prism (HP), rhombic dodecahedron (RD) and truncated octahedron (TO)) is adequately covered in the previous sections. However, it can be summarized that CB, HP and RD model respectively require 85.9, 43.25 and 43.25 % more backbone nodes than the TO model. In order to keep any two physically neighboring backbone nodes within the value of r_{bb}—the maximum separation possible between two backbone nodes for communication link to still operate—CB, HP, RD and TO model requires that r_{bb}/r_{bs} is at least 1.1547, 1.4142, 1.4142 and 1.7889 respectively. Therefore the radius of circumsphere is adjusted accordingly to ensure communication between backbone-backbone and backbone-sensor nodes. When using adjusted TO for $r_{bb}/r_{bs} > 1.7889$, ensuring full coverage automatically realizes that all backbone nodes are connected to all 14 of its physical neighbors. However, when r_{bb}/r_{bs} is small 14-connectivity might not be realized. Relaxing full connectivity with all first tier neighboring nodes makes sense when the nodes are expensive and robust

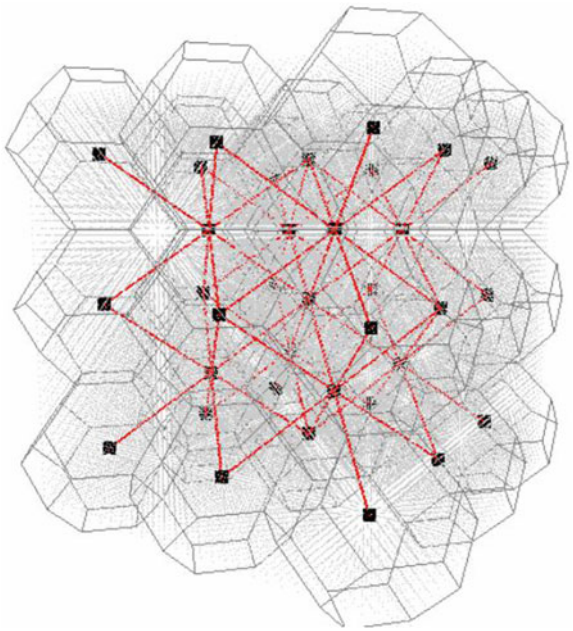


Fig. 20 A TO-based hierarchical network when $1.549 < r_{bb}/r_{bs} < 1.7899$. *Small grey dots* are failure prone mobile sensors. *Large black dots* are backbone nodes. Backbone network links are shown with *red lines*. When $1.7889 < r_{bb}/r_{bs}$, each inner backbone node has 14 links as opposed to 8 links shown in the figure

with very low probability of failure. For example when $1.549 < r_{bb}/r_{bs} < 1.7899$, the backbone node still has connectivity with 8 neighboring backbone nodes which may be sufficient in most applications. Refer to Fig. 20.

8.5 Effects of Oceanic Parameters

It is paramount to have reasonable understanding of behavior and characteristics of acoustic signals when designing 3D submerged network. Naik et al. [44] describes the effect of ocean parameters on acoustic signals in the context node location estimation. Due to acoustic nature of the communication signals, the distance measurements in UW-ASNs can be affected by temperature, salinity and depth. Leroy et al. [42] proposed an equation for sound speed which is a function of temperature (T), salinity (S), depth (Z) and latitude (θ) in all oceans and open seas.

Speed of Sound 'C' $= 1402.5 + 5\mathrm{T} - 5.44\mathrm{x}10^{-2}\mathrm{T}^2 + 2.1\mathrm{x}10^{-4}\mathrm{T}^3 + 1.33\mathrm{S} - 1.23\mathrm{x}10^{-2}\mathrm{ST} + 8.7\mathrm{x}10^{-5}\mathrm{ST}^2 + 1.56\mathrm{x}10^{-2}\mathrm{Z} + 2.55\mathrm{x}10^{-7}\mathrm{Z}^2 - 7.3\mathrm{x}10^{-12}\mathrm{Z}^3 + 1.2\mathrm{x}10^{-6}\mathrm{Z}(\theta\text{-}45) - 9.5\mathrm{x}10^{-13}\mathrm{TZ}^3 + 3\mathrm{x}10^{-7}\mathrm{T}^2\mathrm{Z} + 1.43\mathrm{x}10^{-5}\mathrm{SZ}$.

Due to special mechanical properties of sea water, sound moves at the mean speed of 1500 m/s. It has to be said that network designers should keep these oceanic parameters in mind while deciding on network latency, connectivity, and deployment and configuration algorithms.

9 Surface Coverage in Wireless Sensor Networks

In surface coverage, the target field of interest is a complex surface in 3D space and sensors can be deployed only on the surface. There can be scenarios in many real-world applications where Field of Interest (FoI) is neither a 2D ideal plane nor a full 3D space. Instead they are complex surfaces. For example, in Tungurahua volcano monitoring project [36], 2D plane coverage strategy cannot be used due to the complex surface of the volcano. Similarly, the sensor network cannot be modeled as 3D network since the nodes can only be placed on the surface of the volcano. Zhao et al. [75] addresses the problem of surface coverage by deriving analytical expressions of the expected coverage ratio on surface coverage for stochastic deployment. The work in [75] is compactly summarized in following text.

It is assumed that sensing field is a convex surface which can be modeled as a single valued functions z = *f*(x,y). For stochastic deployment, two sensor distribution models are considered: space surface Poisson point process model and planar surface Poisson point process model. The complex surface is simplified into many small triangles as it pertains to 2D ideal planes to complex surface. An approximate value for the coverage ratio when sensors are stochastically deployed is then achieved.

Jin et al. [40] addresses the surface deployment problem in terms of sensing quality by introducing a general function to measure the unreliability of monitored data in the entire sensor network. Sensors do not always make perfect measurements, but exhibit unreliability that in general depends on the distance between the sensor and target to be sensed.

Let p_i denote the position of sensor i on the Field of Interest denoted as A. Given a point q on A, the sensing unreliability function $g(||q - p_i||)$ describes how unreliable the measurement of information at point q is, sensed by sensor i.

It is demonstrated that centroidal voronoi partition-based sensing regions achieve the optimal solution. Surface parametrization is used for one-to-one mapping from 3D surface to planar domain. However, it is essential to compensate for the distance distortion when computing the generalized centroidal Voronoi Partition on mapped planar disk. The conformal mapping preserves the surface Riemannian metric (distance) up to a scaling factor called conformal factor. The conformal factor cf at a point p on the surface can be computed as ratio between infinitesimal areas around p in the 3D surface and the 2D mapped plane, i.e. $cf(p) = (\text{Area}_{3D}(p))/(\text{Area}_{2D}(p))$. Figure 21 shows the major steps of the algorithm.

In the simulation study [40] sensors are deployed on the testing surface models. The first row of Fig. 22 shows sensors are randomly deployed on the surfaces which give a very high sensing unreliability as shown in Fig. 23. Partitioning the sensing area using voronoi partition of the set of sensors improves the coverage by decreasing sensing unreliability by 47.58 %. However, generalized centroidal voronoi partition-based deployment of sensor nodes yields the optimum solution by decreasing the total sensing unreliability by 89.94 % as compared to random sensing deployment.

It is shown that optimal surface deployment of complex surfaces can be achieved by using centroidal Voronoi-based sensing partition of the Field of Interest.

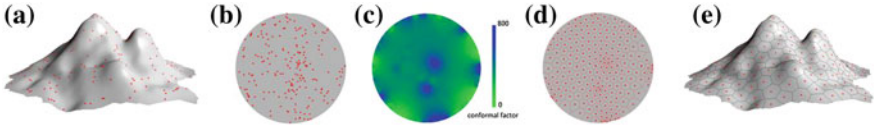


Fig. 21 (Step 1) The algorithm starts with initial random deployment of set of sensors on mountain surface approximated by 5 k *triangles*, denoted as M in **a**. (Step 2) The mountain surface is mapped to a planar unit disk based on a conformal parametrization denoted as f, with sensors mapped to the disk accordingly. **b** shows the computed conformal mapping of the surface to unit planar disk, denoted as $f(M) = D$. (Step 3) Metric distortion of the surface on the disk is measured by conformal factor *cf* , color coded. **c** shows the color encoded *cf,* which measures the metric distortion of M on D. (Step 4) A generalized centroidal Voronoi partition of the set of sensors is computed on the planar disk based on its compensated metric, where points and polygons representing the computed sensor positions and their sensing regions respectively. Computed generalized centroidal Voronoi partition on D based on its compensated metric is shown in **d** with *red points*and marked polygons representing the computed sensor positions and their sensing regions respectively. (Step 5) The set of sensors and their corresponding sensing regions are projected back to the surface. **e** depicts the optimal deployment of the set of sensors and their sensing regions on M by projecting the computed generalized centroidal Voronoi partition on D to M based on f^{-1} [40]

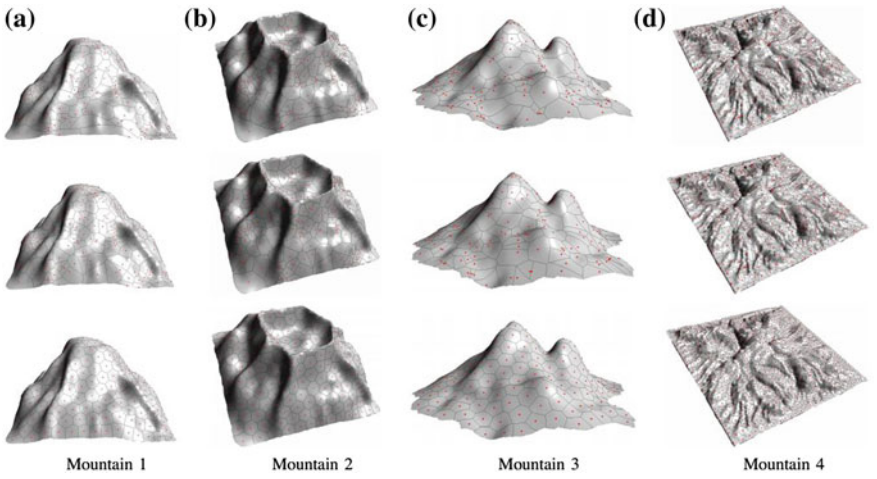


Fig. 22 The *first row* shows random sensing partition. The *second row* shows Voronoi-based sensing partition, and the *third row* shows generalized centroidal voronoi-based sensing partition [40]

The scheme uses conformal mapping to estimate the 3D depth of the region and then partitions the field accordingly for optimal sensor deployment resulting in minimum sensing unreliability of the Field of Interest.

Liu et al. [43] study the coverage problem for rolling terrains, and derive the expected coverage ratios under the stochastic sensors deployment. Both regular and irregular terrain coverage problems are analyzed. For regular terrains, general expression of the expected coverage ratio for the 3D surface function $z = f(x,y)$ is derived. For irregular terrains, Digital Elevation Model (DEM) is used to derive analytical

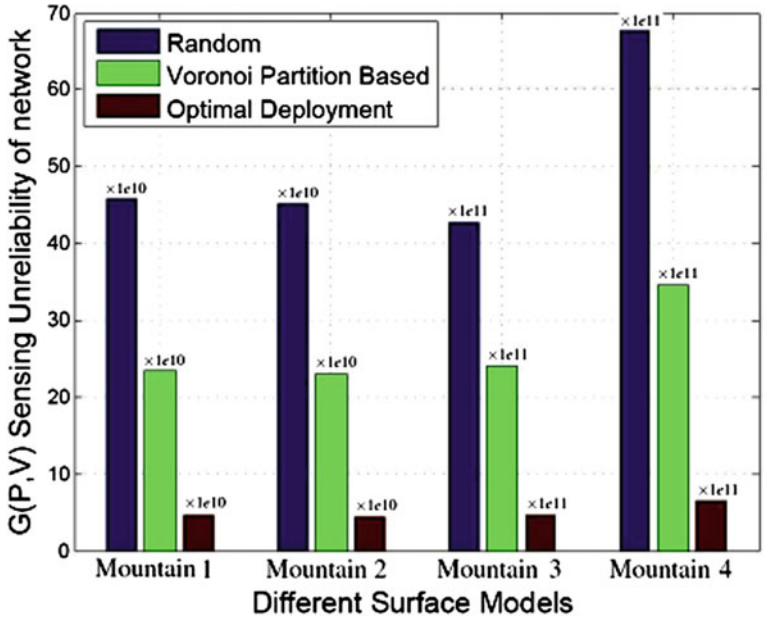


Fig. 23 Sensing unreliability with different deployment and sensing partition methods on various surface models as shown in Fig. 22 [40]

expression of the expected coverage ratios. By partitioning the Field of Interest in areas of small triangles which have very have very small dimensions compared to the sensing radius of the sensor, the coverage ratio of the surface is estimated. A triangle is considered covered only if all its vertexes are covered. The ratio between the sum of covered triangle's area to the total surface area yields the coverage ratio.

10 Mobility

The mobility of sensor nodes can be exploited to enhance coverage of the network. The objective is to determine positions and/or movements of nodes to achieve maximum coverage and to form a uniformly distributed wireless network. As opposed to static nodes, the support of mobility in a network brings flexibility as well as adaptability. It gives the network designers the convenience to change the location of nodes depending on the application time specific requirements.

Fang et al. [27] propose a set of algorithms for coverage enhancement in mobility enabled sensor network. Fang et al. use the idea of virtual force proposed in [76] to calculate the net virtual force which dictates the direction of the movement of mobile sensor node. These forces can be either positive (attractive) or negative (repulsive). The movement of a sensor s_i will be calculated as a vector summation of all forces

executed on s_i. The solution proposed in [27] is listed below. This work can be extended for many mobility-based coverage and connectivity applications.

10.1 Type of Virtual Forces

10.1.1 Attractive Force of Target Field

When a sensor s_i is place outside of region ζ_t, ζ_t will execute attractive force on s_i denoted by $\mathrm{F_A}(s_i)$, which is defined as

$$.\mathbf{FA}(si) = \begin{cases} l_{\mathrm{A}} - l_i & \text{if } l_i \text{ is not member of } \zeta_{\mathrm{t}} \\ 0 & \text{otherwise} \end{cases}$$

where A is the point in ζ_t which is nearest to s_i. l_A represents location of A in cartesian coordinates. $l_A - l_i$ is the vector which points from location of $s_i\,(\mathrm{x_i}, \mathrm{y_i}, \mathrm{z_i})$ to location of A $(\mathrm{x_A}, \mathrm{y_A}, \mathrm{z_A})$.

10.1.2 Repulsive Force of Boundary

When a sensor s_i is placed in region ζ_{t}, and is close to boundary of ζ_{t},the boundary exert repulsive force on s_i denoted by $\mathrm{F_B}(s_i)$. When ζ_{t} is in 3D space, the boundary will be a 2D surface. The boundary (extended boundary) is visualized as 2D space which is perpendicular to ζ_{t} and contains boundary of ζ_{t}. It will be some planes which are perpendicular to ζ_{t} and include the corresponding line segments Fig. 24.

$$\mathbf{FB}\,(si) = \begin{cases} \frac{l_i - l_B}{\mathrm{d(B,\,s_i)}} \cdot (\mathrm{r} - \mathrm{d\,(B,s_i)}) & \text{if } l_i \in \zeta\mathrm{t} - \mathrm{B\,AND\,d\,(B,s_i)} < r \\ \nu.\mathrm{r} & \text{if } l_i = \mathrm{B} \\ 0 & \text{otherwise} \end{cases}$$

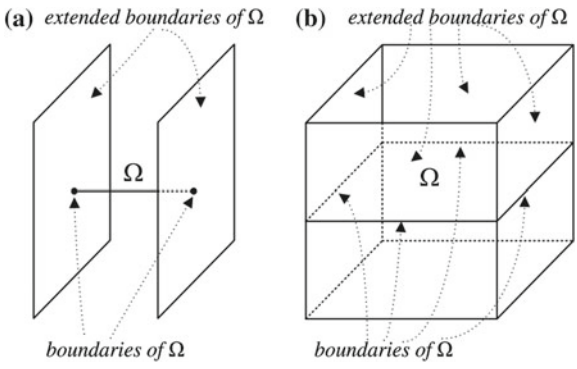


Fig. 24 The extended boundary concept as illustrated in [27]

If there are m boundaries having distance less than r from s_i, then

$$\mathbf{FB}\,(si) = \sum_{\mathrm{j}=1}^{\mathrm{m}} \mathbf{FB}\,(si)\; j$$

Each sensor checks its location and if it finds itself too close to boundaries, it will bounce back by the resultant force from these boundaries.

10.1.3 Repulsive Force Between Sensors

To maintain optimum separation between the sensors, the sensors also exert repulsive force on each other defined by

$$\mathbf{FC}\,(si, sj) = \begin{cases} \frac{l_i - l_j}{\mathrm{d}(\mathrm{s_i},\mathrm{s_j})} \cdot \left(2\mathrm{r} - \mathrm{d}\left(\mathrm{s_i},\mathrm{s_j}\right)\right) \text{ if } \mathrm{d}\left(\mathrm{s_i},\mathrm{s_j}\right) < 2r \\ 0 \qquad\qquad\qquad\qquad \text{otherwise} \end{cases}$$

The total repulsive force on a sensor is the vector summation of repulsive forces exerted by all the other sensors.

10.1.4 Attractive Force of Potential Field

The most important challenge for moving a sensor to a potential field or repairing a coverage hole is to detect the potential field or coverage hole effectively and accurately. A sensor is moved to uncovered area if it is located adjacent to such an uncovered patch. Based on spherical overlap coverage detection method in [37], the adjacent potential field coverage is detected for a given sensor s_i.

As shown in Fig. 25, for $ARC_j^i q$ which is part of *Cir(i,j),* there exists an adjacent potential field ζjq which includes $ARC_j^i q$, if $ARC_j^i q$ is not covered by any other sensor besides s_i and s_j. The attractive force exerted by ζjq on s_i is defined as

$$\mathbf{FD}\,(si, \mathrm{1}jq) = \begin{cases} \frac{l_{\mathrm{pq}} - l_{oij}}{\mathrm{d}(s_i, s_j)} \cdot \sin\left(\frac{\theta}{2}\right) \text{ ifCe } (s_\mathrm{i}, \mathrm{1_t}) \text{ is not equal} \\ 0 \qquad\qquad\qquad\qquad\qquad\qquad \text{otherwise} \end{cases}$$

where Oij is center point of *Cir(i,j)*, is the central angle for $ARC_j^i q$, Pq denote the midpoint of this arc. The total attractive force on a sensor is the vector summation of all forces exerted by adjacent potential fields.

10.2 Movement of Sensors

The total virtual force exerted on the sensor is calculated by the weighted addition of all the four types of forces. The resultant movement $\boldsymbol{m}_i$ is expressed as

$\boldsymbol{m}_i = C_A.\mathbf{FA}(\mathrm{s_i}) + C_B.\mathbf{FB}(\mathrm{s_i}) + C_C.\mathbf{FC}(\mathrm{s_i}) + C_D.\mathbf{FD}(\mathrm{s_i})$, where C_A, C_B, C_C and C_D are constant factors [27].

11 Major Work Summary

This section lists the major works on Coverage and Connectivity in 3D WSNs. The Table 11 categorizes the work based on lattice pattern, placement strategy, major contribution, and relevant application.

12 Open Problems and Research Challenges

This section highlights the major challenges and open problems for future research work on Coverage and Connectivity in 3D WSNs.

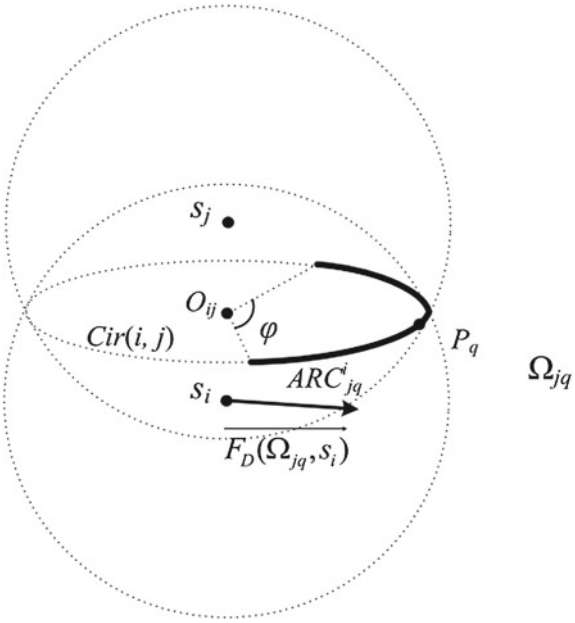


Fig. 25 Attractive force of 3D potential field as illustrated in [27]

Table 11 Major Works on Coverage and Connectivity in 3D WSNs

#	Placement / Strategy	Model / Base	Relevant / Applica-tion	Contribution	Network	Mobility	Lattice	Solution for	Limiting / Factor	Notes
[4, 6–8]	Deterministic	Voronoi tessellation	General placement	Optimal placement strategies	Homogeneous	No	RD, HP, TO, CB	Maximum coverage	Rc/Rs	Chooses optimum lattice for different ranges of Rc/Rs
[38, 39, 62]	Deterministic	Voronoi tessellation	General placement	Benefit of TO in terms of energy efficiency	Homogeneous	No	TO	Maximum coverage	Rc/Rs	Active-idle scheme for increased lifespan in TO lattice
[22]	Deterministic	Voronoi tessellation	General placement	BCC lattice comparison	Homogeneous	No	BCC	Maximum coverage	1-coverage	Active-idle scheme for increased lifespan in BCC lattice
[9, 10]	Deterministic	Voronoi tessellation	General Placement	Deployment based on Reuleaux tetrahedron	Homogeneous	No	RT	k-coverage	nodal distance	Achieve 4-coverage using RT model
[9, 26]	Deterministic	Voronoi tessellation	General placement	Conditional connectivity	Both	No**	RT in [9]	k-connectivity	Faulty Node	Increased network resilience by including faulty set in model
[21, 37]	Not relevant	Spherical overlap	Coverage analysis	estimate k-coverage of sphere	Homogeneous*	No**	No	estimate k-coverage	NA	Uses intersection of spheres to estimate coverage degree

(continued)

Table 11 (Continued)

#	Placement Strategy	Model Base	Relevant Application	Contribution	Network	Mobility	Lattice	Solution for	Limiting Factor	Notes
[15]	Deterministic	Voronoi tessellation	General placement	Placement analysis of full coverage and high connectivity	Homogeneous	No	CB	High connectivity k = 6, 14	Rc/Rs	Lattice Patterns for 6-,14-connectivity
[51]	Random	Spatial node distribution	threshold for coverage and connectivity	critical transmission and sensing ranges for degree of coverage and connectivity	Homogeneous	No	No	coverage, connectivity Hop Diameter	Rc/Rs	critical Rc/Rs ranges for various degrees of connectivity, coverage, hop diameter
[16, 74]	Deterministic	Voronoi tessellation	General placement	Placement analysis of full coverage and Low connectivity	Homogeneous	No	CB	Low connectivity k< = 4	Rc/Rs	Lattice Patterns for 1-,2-,3-,4-Connectivity
[8]	Deterministic	Strip based	General placement	Full coverage and 1 connectivity	Homogeneous	No	No	coverage	connectivity	Scheme is simple and less computationally intensive
[12]	Random	Continuum percolation	General placement	nodal critical density for percolation in coverage and connectivity	Both	No**	No	node critical density	overlap sensing volume	Gives integrated continuum percolation for coverage and connectivity

(continued)

Table 11 (Continued)

#	Placement Strategy	Model Base	Relevant Application	Contribution	Network	Mobility	Lattice	Solution for	Limiting Factor	Notes
[48]	Random, Bottom random, Bottom grid	No special scheme	Submerged networks	1-coverage in UW-ASN	Homogeneous	No	No	1-coverage	Practical issues	Generic placement scheme for UW-ASN
[19, 57]	Grid based	Cubicle coordinate system	Submerged networks	1-coverage in UW-ASN (Maximized)	Homogeneous	No	No	1-coverage	Practical issues	Nodes coordinate with one another to achieve maximum separation and full coverage
[2]	Random, Self configuration	Clustering	Submerged Networks	1-coverage in UW-ASN (Maximized)	Homogeneous*	No**	CO	1-coverage	Practical Issues	Four stage self organization: Clustering, grouping, depth assignment, iteration rounds
[5]	Deterministic	Voronoi tessellation	Submerged networks	Maximum coverage	Heterogeneous Hierarchical	No	RD, HP, TO, CB	Maximum coverage	Rc, Rs, Rbc, Rbs	Incorporates concept of backbone nodes in 3D WSN Analysis
[75]	Stochastic Deployment	Voronoi tessellation	Surface coverage	Approximation of coverage ratio for 3D surfaces	Homogeneous	No	No	estimate k-coverage	NA	Estimates coverage by resolving complex 3D surface into 2D surface

(continued)

Table 11 (Continued)

#	Placement Strategy	Model Base	Relevant Application	Contribution	Network	Mobility	Lattice	Solution for	Limiting Factor	Notes
[40]	Random vs Deterministic	Voronoi partition, Surface parameterization	Surface coverage	Voronoi partition of 3D surface for node place resulting in improved coverage	Homogeneous	No	No	Maximum coverage	No of nodes	Uses conformal mapping to account for surface distortion
[43]	Stochastic Deployment	Voronoi partition	Surface coverage (Rolling Terrains)	Voronoi partition of 3D surface for node place resulting in improved coverage	Homogeneous	No	No	Maximum coverage	No of nodes	Uses sensing unreliability function to derive analytical expression of expected coverage ratio
[27]	Random, Self configuration	Virtual forces	General placement	Coverage enhancement in mobility enabled WSN	Homogeneous	Yes	No	Maximum coverage	Coverage connectivity	Virtual forces act on sensor node. The node moves in direction of resultant force

Legend
*Model can be extended for heterogeneous networks
**Model can extended for Mobility
Rc/Rs: Ratio of Communication (Transmission) Radius to Sensing Radius
Rbc: Backbone Node Communication Radius
Rbs: Backbone node Sensing Radius
RD: Rhombic Dodecahedron
TO: Truncated Octahedron
CB: Cuboid Lattice
HP: Hexagonal Prism
CO: Centralized Octahedron

12.1 Mobility

Though some work has been done in the domain of Mobility in 3D WSNs, still there is vast scope for future work in this field. Many perspective applications for submerged networks, including tactical deployments and surveillance, may require mobility supported WSNs. Coordinated operations of Mobile and Static Sensing Nodes in unfriendly environment is a major design challenge. Furthermore most of the present work visualizes sensor nodes as small mobile platforms highly efficient in movement. This assumption is not practical. AUVs are usually slow and bulky machines. Furthermore, the number of nodes is also limited. Mobility model for 3D WSNs which addresses practical constraints is a major open problem in this field.

12.2 Heterogeneous Sensor Nodes

Most present work assumes the sensor nodes to be homogeneous in terms of computational ability, sensing and communication radii, life span etc. However this is not true in most applications. In some applications different classes of nodes may be deployed. While in other applications same types/classes of nodes might behave differently. For example, it is a common assumption to consider sensing and communication radii of nodes to be same network wide. Many placement strategies use Rc/Rs to find optimal 3D node placement strategies. However even if the nodes have same hardware, their Rc/Rs might be considerably different due to non-uniform sensor workload and power dissipation and local environmental characteristics. In most cases, the sensing range is location dependent and is highly irregular. Multi-path and shadowing dramatically affects the communication range of a sensor, sometimes with nodes nearer to each other have no links while nodes apparently far distant from each other have strong links. More realistic models, which includes practical constraints and considers nodes as heterogeneous as opposed to homogeneous are required.

12.3 Heterogeneous Networks

Although in much of present literature, the term heterogeneous networks usually refers to presence of nodes with non-uniform sensing and communication radii in the networks and it is used interchangeably with heterogeneous nodes, the requirement for considering the whole network heterogeneous in terms of regional nodal densities, propagation characteristics, latency, coverage and connectivity requirements is an unexplored research problem. A network can be considered heterogeneous if its regional characteristics and requirements are not same as universal. For example some applications might require higher degree of coverage and connectivity in certain

regions of the networks as opposed to others. One such work is [40]. It would be an interesting work to incorporate 'location dependent value' in the deployment model to assign different node densities to different regions in the monitored zone according to requirements. Models for such networks hardly exist at the moment. It would be remarkable work to formulate simple 3D node placement models which addresses the heterogeneous coverage and connectivity requirements of 3D WSNs.

12.4 Coverage in Obstacles

Coverage in the presence of obstacles is a challenging problem and has not been addressed much in the literature. For example, in many pre and post disaster management applications, it is required to operate in hostile, obstacle prone and unpredictable environments. Studying of propagation characteristics both for electromagnetic and acoustic signals specific to 3D WSNs would be a major step forward in enhancing coverage in obstacle prone 3D fields. Modeling Obstacles with arbitrary shapes and improving coverage and connectivity in such spaces is an open problem and existing tools and techniques need to be substantially extended to meet these challenges.

12.5 Node Retrieval and Failure Analysis

While there are many optimal placement strategies both random and deterministic, there is almost no node retrieval strategy. In many practical applications, there can be numerous scenarios when a node needs to be replaced or retrieved. Such scenarios are plausible when nodes are expensive and non-expendable, e.g., tactical systems. For such scenarios analysis needs to be done for replacing a node, the degree of acceptable tolerance in position dislocation, its effect on regional coverage and connectivity etc.

13 Conclusion

This work is a survey of recent efforts and major works in the field of Coverage and Connectivity in 3D WSNs. 3D WSNs is an emerging arm of WSNs and has applications in the defense, space exploration, submerged networks and airborne reconnaissance. In this work we have tended to enlist major 3D node placement techniques based on the cell partitioning schemes. Generally speaking, a 3D space can be partitioned using space filling polyhedrons which tile in 3D space. The four major space filling polyhedrons are truncated octahedron (TO), rhombic dodecahedron (RD), cube (CB) and hexagonal prism (HP). Though TO based cell partitioning usually yields optimal solution in terms of number of nodes required, it however requires relatively higher coverage to communication radii ratio. This might make

TO impractical for many scenarios. CB is welcoming alternate which achieves same coverage with slightly increased number of minimum required nodes. Due to simpler model and easier mathematical manipulations, CB based models can also be readily extended to meet different degrees of desired connectivity. By varying the lengths of the edges of cube (cuboid) different mutually interdependent k-connectivity lattice patterns can be derived easily. For k-coverage realueaux tetrahedrons based deployment schemes are used which employ spherical overlap of sensing radii of nodes to yield desired degree of coverage. For random deployments continuity percolation based analysis yield the critical densities for desired degrees of coverage and connectivity. Continuity Percolation based analysis is useful for scenarios involving huge number of nodes dispersed randomly in large monitoring zones. Underwater sensor networks are by far the most common example of 3D WSNs. This work also lists major deployment schemes for submerged networks along with their practical constraints and design challenges. To abridge, this survey enumerates the major techniques for systematic and random node deployments for 3D environments, their comparisons and practical design constraints for such networks.

Acknowledgments The authors gratefully acknowledge the insightful comments of the anonymous reviewers which helped improve the quality and presentation of the paper significantly. This work is partially supported by the US National Science Foundation (NSF) grant 1054935 and 1224628.

References

1. Aerial common sensor (acs) (2007), http://www.globalsecurity.org/intell/systems/acs.htm
2. K. Akkaya, A. Newell, Self-deployment of sensors for maximized coverage in underwater acoustic sensor networks. Comput. Commun. **32**(7–10), 1233–1244, ISSN 0140-3664 (2009). doi:10.1016/j.comcom.2009.04.002
3. I.F. Akyildiz, D. Pompili, T. Melodia, Underwater acoustic sensor networks: research challenges, Ad Hoc Netw. **3**(3), 257–279, ISSN 1570-8705 (2005). doi:10.1016/j.adhoc.2005.01.004
4. S.M.N. Alam, Z.J. Haas, Coverage and connectivity in three-dimensional networks. in *Proceedings of the 12th Annual International Conference on Mobile Computing and Networking (MobiCom '06)*, (ACM, New York, 2006), pp. 346–357. doi:10.1145/1161089.1161128
5. S.M.N. Alam, Z.J. Haas, Hierarchical and nonhierarchical three-dimensional underwater wireless sensor networks CoRR abs/1005.3073: (2010)
6. S.M.N. Alam, Z.J. Haas, Topology control and network lifetime in three-dimensional wireless sensor networks. Presented at CoRR, (2006)
7. S.M.N. Alam, Z.J. Haas, Topology control of three-dimensional underwater wireless sensor networks. Underw. Acoust. Sens. Netw. 45–69 (2010)
8. S.M.N. Alam, Z.J. Haas, Coverage and connectivity in three-dimensional underwater sensor networks. Wirel. Commun. Mob. Comput. **8**, 995–1009 (2008). doi:10.1002/wcm.661
9. H.M. Ammari, S.K. Das, A study of k-coverage and measures of connectivity in 3D wireless sensor networks. IEEE Trans. Comput. **59**(2), 243–257 (2010). doi:10.1109/TC.2009.166
10. H.M. Ammari, S.K. Das, Joint k-coverage and hybrid forwarding in duty-cycled three-dimensional wireless sensor networks. 5th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks, SECON '08. pp. 170–178, (2008). doi:10.1109/SAHCN.2008.30

11. H. M. Ammari, S. K. Das, Lecture Notes in Computer Science (LNCS). in *Proceedings EWSN. Clustering-Based Minimum Energy m-Connected k-Covered Wireless Sensor Networks***4913**, pp. 1–16 (2008)
12. H.M. Ammari, S.K. Das, Critical density for coverage and connectivity in three-dimensional wireless sensor networks using continuum percolation. IEEE Trans. Parallel Distrib. Syst. **20**(6), 872–885 (2009). doi:10.1109/TPDS.2008.146
13. AUV Laboratory at MIT Sea Grant, http://auvlab.mit.edu/
14. X. Bai, S. Kumar, Z. Yun, D. Xuan, T.H. Lai, Deploying wireless sensors to achieve both coverage and connectivity. in *Proceedings of ACM MobiHoc* (2006)
15. X. Bai, C. Zhang, D. Xuan, J. Teng, W. Jia, Full-coverage and k-connectivity (k=14,6) three dimensional networks. IEEE INFOCOM 2009, pp. 388–396 (2009). doi:10.1109/INFCOM.2009.5061943
16. X. Bai, C. Zhang, D. Xuan, J. Teng, W. Jia, Low-connectivity and full-coverage three dimensional wireless sensor networks. in *Proceedings of the tenth ACM international symposium on Mobile ad hoc networking and computing (MobiHoc '09)*, (ACM, New York, 2009), pp. 145–154. http://doi.acm.org/10.1145/1530748.1530768
17. S.A. Borbash, E.H. Jennings, Distributed topology control algorithm for multihop, wireless networks. in *Proceedings of the IEEE International Joint Conference on Neural Networks* pp. 355–360 (2002)
18. M. Campbell, Intelligence in three dimensions: we live in a 3-d world, and so should computers (2006), http://www.neptec.com/News2006/1Oct06-MilAero.html
19. E. Cayirci, H. Tezcan, Y. Dogan, V. Coskun, Wireless sensor networks for underwater surveillance systems. Ad Hoc Netw. **4**(4), 431–446, ISSN 1570-8705 (2006). doi:10.1016/j.adhoc.2004.10.008
20. C.-Y. Chong, S.P. Kumar, Sensor networks: evolution, opportunities, and challenges. IEEE Proc. **91**(8), 1247–1256 (2003). doi:10.1109/JPROC.2003.814918
21. C-F. Huang, Y-C. Tseng, L-C. Lo, The coverage problem in three-dimensional wireless sensor networks. IEEE Global Telecommunications Conference, 2004. GLOBECOM '04. **5**, 3182–3186 (2004). doi:10.1109/GLOCOM.2004.1378938
22. Sesh Commuri, Mohamed K. Watfa, Coverage strategies in wireless sensor networks. Int. J. Distrib. Sens. Netw. **2**(4), 333–353 (2006). doi:10.1080/15501320600719151
23. J. Conway, S. Torquato, Tiling, packing, and covering with tetrahedra. Proc. Nat.l Acad. Sci. USA. (PNAS) **103**(28), 10612–10617 (2006)
24. K. Day, The conditional node connectivity of the k-ary n-cube. J. Int. Netw. **5**(1), 13–26 (2004)
25. Encyclopedia Geometrica. 3D Reueaux. http://www.fastgeometry.com/Reuleaux/ConstantBreadth3DShapes.htm
26. A. Esfahanian, Generalized measures of fault tolerance with application to n-cube networks. IEEE Trans. Comput. **38**(11), 1586–1591 (1989)
27. C. Fang, P. Zhang, C. Fu, Z. Zhang, Coverage enhancement by using the mobility of mobile sensor nodes. Springer Science+Business Media, LLC (2012). doi:10.1007/s11042-012-1139-4
28. M. Gardner, *The Sixth Book of Mathematical Games from Scientific American* (University of Chicago Press, Chicago, 1984)
29. A. Ghosh, S.K. Das, Coverage and connectivity issues in wireless sensor networks: a survey. Pervasive Mobile Comput. **4**(3), 303–334 (2008)
30. E.N. Gilbert, Random Plane Networks. J. SIAM **9**(4), 533–543 (1961)
31. A. Goel, S. Rai, B. Krishnamachari, Proccedings of the Thirty-Sixth Annual ACM Symposium on Theory of Computing (STOC), pp. 580–586, Chicago, Illinois, June 13–15 (2004)
32. B. Grünbaum, G.C. Shephard, Tilings with Congruent Tiles. Bull. Amer. Math. Soc. **3**, 951–973 (1980)
33. P. Gupta, P.R. Kumar, Internet in the sky: the capacity of three dimensional wireless networks. Comm. Inf. Syst. **1**, 33–49 (2001)
34. P. Hall, On continuum percolation. Ann. Probab. **13**(4), 1250–1266 (1985)
35. F. Harary, Conditional Connectivity. Networks **13**, 347–357 (1983)

36. http://fiji.eecs.harvard.edu/Volcano/
37. C. Huang, Y. Tseng, L. Lo, The coverage problem in three-dimensional wireless sensor networks. J. Int. Netw. **8**(3), 209–227 (2007)
38. A. Jawahar, S. Radha, S. Vadivelan, Connectivity-guaranteed hybrid topology management scheme for improving the operational lifetime of 3-dimensional wireless sensor networks. Int. J. Distrib. Sens. Netw. **2010**, Article ID 547368, 11p (2010). doi:10.1155/2010/547368
39. Z. Q. Jiang et al., Deployment with sampling coverage in three-dimensional wireless sensor networks. Appl. Mech. Mater. **43**, 342 (2010). doi:10.4028/www.scientific.net/AMM.43.342
40. M. Jin, G. Rong, H. Wu, L. Shuai, X. Guo, Optimal surface deployment problem in wireless sensor networks. in *Proceedings of IEEE INFOCOM 2012*. pp. 2345–2353 (2012). doi:10.1109/INFCOM.2012.6195622
41. N.W. Johnson, *Uniform Polytopes* (Cambridge University Press, Cambridge, 2000)
42. C.C. Leroy, P. Stephen, A new equation for the accurate calculation of sound speed in all Oceans. J. acoust. Soc. Am. **124**(5), 2774–2782 (2008)
43. L. Liu, H. Ma, On coverage of wireless sensor networks for rolling terrains. IEEE Trans. Parallel Distrib. Syst. **23**(1), 118–125 (2012). doi:10.1109/TPDS.2011.69
44. S.S. Naik, M.J. Nene, Realization of 3D underwater wireless sensor networks and influence of ocean parameters on node location estimation. Int. J. Wirel. Mob. Netw. **4**(2), 135 (2012)
45. Ocean engineering at Florida Atlantic university, http://www.oe.fau.edu/research/ams.html
46. J. Partan, J. Kurose, N. Levine, A survey of practical issues in underwater networks WUWNet (2006)
47. S. Poduri, S. Pattem, B. Krishnamachari, G.S. Sukhatme, Sensor Network Configuration and the Curse of Dimensionality. in *Proceedings of Third IEEE Workshop Embedded Networked Sensors (EmNets)* (2006)
48. D. Pompili, T. Melodia, I.F. Akyildiz, Deployment analysis in underwater acoustic wireless sensor networks. in *Proceedings of the 1st ACM International Workshop on Underwater Networks (WUWNet '06)*, (ACM, New York, 2006), pp. 48–55
49. D. Pompili, T. Melodia, I. F. Akyildiz, Three-dimensional and two-dimensional deployment analysis for underwater acoustic sensor networks. Ad Hoc Netw. **7**(4), 778–790, ISSN 1570–8705 (2009). doi:10.1016/j.adhoc.2008.07.010
50. J. Proakis, E. Sozer, J. Rice, M. Stojanovic, Shallow water acoustic networks. IEEE Commun. Mag. **39**(11), 114–119 (2001)
51. V. Ravelomanana, Extremal properties of three-dimensional sensor networks with applications. IEEE Trans. Mob. Comput. **3**(3), 246–257 (2004). doi:10.1109/TMC.2004.23
52. P. Santi, D.M. Blough, The critical transmitting range for connectivity in sparse wireless ad hoc networks. IEEE Trans. Mob. Comput. **2**(1), 25–39 (2003)
53. S. Shakkottai, R. Srikant, N. Shroff, Unreliable sensor grids: coverage, connectivity and diameter, Ad Hoc Networks
54. A. Goel, S. Rai, B. Krishnamachari, Monotone properties of random geometric graphs thresholds. Ann. Appl. Probab. **15**(4), 2535–2552 (2005)
55. H. Steinhaus, *Mathematical Snapshots*, 3rd edn. (Oxford University Press, New York, 1969)
56. M. Stojanovic, *acoustic (underwater) Communications*, ed. by J.G. Proakis. Encyclopedia of Telecommunications, (John Wiley and Sons, Chichester, 2003)
57. H. Tezcan, E. Cayirci, V. Coskun, A distributed scheme for 3D space coverage in tactical underwater sensor networks. in , *IEEE Military Communications Conference MILCOM 2004*, **2**, pp. 697–703 (2004). doi:10.1109/MILCOM.2004.1494881
58. S. Torquato, Y. Jiao, Dense packings of polyhedra: platonic and archimedean solids Physical Review E - Statistical. Nonlinear Soft Matter Phys. **80**(4), 041104 (2009). doi:10.1103/PhysRevE.80.041104
59. W. Tsujita, A. Yoshino, H. Ishida, T. Moriizumi, Gas sensor network for air-pollution monitoring. Sens. Actuat. B-Chem. **110**(2), 304–311 (2005)
60. UnderWater Sensor Networks at BWN Laboratory, Georgia institute of technology, http://www.ece.gatech.edu/research/labs/bwn/UWASN/

61. K. Vasilescu, D. Kotay, M. Rus, P. Dunbabin, Corke, data collection, storage, and retrieval with an underwater sensor network, in SenSys '05. in *Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems*, (ACM Press, New York, 2005), pp. 154–165
62. X. Wang, J. Wu, L. Guo, A k-coverage algorithm in three dimensional wireless sensor networks. in *3rd IEEE International Conference on Broadband Network and Multimedia Technology (IC-BNMT)*, pp. 1089–1093 (2010). doi:10.1109/ICBNMT.2010.5705257
63. M.K. Watfa, S. Commuri, Optimal 3-dimensional sensor deployment strategy. in *3rd IEEE Consumer Communications and Networking Conference, 2006*, **2**, pp. 892–896 (2006). doi:10.1109/CCNC.2006.1593167
64. M.K. Watfa, S. Commuri, The 3-Dimensional wireless sensor network coverage problem. in *Proceedings of the 2006 IEEE International Conference on Networking, Sensing and Control ICNSC '06*, pp. 856–861(2006). doi:10.1109/ICNSC.2006.1673259
65. M.K. Watfa, Practical applications and connectivity: algorithms in future wireless sensor networks. Int. J. Inf. Technol. **4**, 18–28 (2007)
66. R. Wattenhofer, L. Li, P. Bahl, Y.M. Wang, A conebased distributed topology-control algorithm for wireless multi-hop networks. IEEE/ACM Trans. Netw. **13**(1), 147–159 (2005)
67. Weisstein, W. Eric , Open ball. From mathworld-A wolfram web resource. http://mathworld.wolfram.com/OpenBall.html
68. Weisstein, W. Eric, Reuleaux tetrahedron. From mathworld-A wolfram web resource. http://mathworld.wolfram.com/ReuleauxTetrahedron.html
69. Weisstein, W. Eric, Space-filling polyhedron."From mathworld-A wolfram web resource. http://mathworld.wolfram.com/Space-FillingPolyhedron.htm
70. D. Wells, *The Penguin Dictionary of Curious and Interesting Geometry* (Penguin, London, 1991)
71. Z. Xiao, M. Huang, J. Shi, J. Yang, J. Peng, Full connectivity and probabilistic coverage in random deployed 3D WSNs. International conference on wireless communications and, signal processing, pp. 1–4 (2009). doi:10.1109/WCSP.2009.5371666
72. X. Yang, K.G. Ong, W.R. Dreschel, K. Zeng, C.S. Mungle, C.A. Grimes, Design of a wireless sensor network for long-term, in-situ monitoring of aquatic environments. Sensors **2**, 455–472 (2002)
73. M. Younis, K. Akkaya, Strategies and techniques for node placement in wireless sensor networks: a survey, Ad Hoc Networks **6**(4), pp. 621–655, ISSN 1570–8705 (2008). doi:10.1016/j.adhoc.2007.05.003
74. C. Zhang, X. Bai, J. Teng, D. Xuan, W. Jia, Constructing low-connectivity and full-coverage three dimensional sensor networks. IEEE J. Sel. Areas Commun. **28**(7), 984–993 (2010). doi:10.1109/JSAC.2010.100903
75. M.-C. Zhao, J. Lei, M.-Y. Wu, Y. Liu, W. Shu, Surface coverage in wireless sensor networks IEEE INFOCOM **2009**, 109–117 (2009). doi:10.1109/INFCOM.2009.5061912
76. Y. Zou, K. Chakrabarty, Sensor deployment and target localization in distributed sensor networks. ACM Trans. Embed. Comput. Syst. (TECS) **3**(1), 61–69 (2004)

Chapter 9
Localization in Three-Dimensional Wireless Sensor Networks

Usman Mansoor and Habib M. Ammari

Abstract A wireless sensor network (WSN) is categorized as three-dimensional (3D) when the variation in the height of deployed sensor nodes is not negligible as compared to length and breadth of deployment field. Localization is one of the fundamental components of any wireless sensor application. A localization algorithm estimates the position of a node by using information provided/inferred from anchor beacons, reference nodes or neighbors connectivity. The effectiveness of a localization algorithm is usually determined in terms of accuracy, resilience to node failure, computational cost, messaging overhead, hardware constraints and deployment practicality. This survey overviews the major recent work done in the field of localization in 3D WSNs. The major contribution of this work is to present all the major 3D (generic, airborne, terrestrial and submerged) localization schemes in a single literature along with their relative strengths and weaknesses.

1 Introduction

Localization in Wireless Sensor Networks is a key enabling technology and takes an essential role in many practical applications. The inherent objective of any localization scheme is to estimate node position with reasonable accuracy (specific to application) with minimal cost (economic, computational, messaging, energy). Accuracy and resilience requirements, prospective deployment environment, hardware constraints, energy conservation targets, and economic viability play a critical role when designing a localization scheme tailored for a certain application [66].

A WSN is categorized as 3D when the variation in height of deployed sensors is not negligible as compared to length and width of the deployment field. Convention-

U. Mansoor · H. M. Ammari (✉)
WiSeMAN Research Lab, CIS Department, University of Michigan—Dearborn, Dearborn, MI, USA
e-mail: hammari@umd.umich.edu

H. M. Ammari (ed.), *The Art of Wireless Sensor Networks*, Signals and Communication Technology, DOI: 10.1007/978-3-642-40066-7_9,

ally, sensor networks are usually visualized as two-dimensional (2D) networks. This assumption is valid for most terrestrial scenarios since nodes are deployed in same planes and there is usually little or no across-planes nodal communication. However, this 2D model loses its relevance for most submerged and airborne deployments [1, 10, 60, 92, 99].

There has been considerable work done in the domain of localization in WSNs complemented by some excellent surveys in the field [3, 28, 30, 31, 45, 51, 71]. Although these works adequately address the localization problem in general, concentrated work in 3D localization for WSNs is still to be undertaken. Interestingly, most present work in 3D localization for WSNs gyrate around theoretical and conceptual designs, and little effort has been extended towards proposing localization schemes based on practical real-world 3D WSNs. It can be said that there is scope for ground breaking research and new ideas for practical 3D localization schemes in Wireless Sensor Networks.

Extensibility of a 2D localization scheme to 3D is an algorithmic challenge rather than a hardware challenge. Any localization systems/scheme can be divided into three distinct components: distance/angle estimation, position computation, and localization algorithm (refer to [6] for more details). Either range based (ToA, TDoA, RSSI) or range free (connectivity, proximity) techniques are employed to estimate/infer distance/angle which is fed into position computation algorithm to compute relative position of the node from anchors. While "distance/angle estimation" and "position computation" deal with the ranging techniques, measurements and location calculation of the nodes, "localization algorithm" outlays the generic operation of the scheme, its hardware requirements and constraints, operational environment (submerged, terrestrial or airborne), anchor types, etc. It is usually designed while considering prospective applications, deployment and operational costs, etc. The distance estimation component usually is not dimension dependent and can be easily extended for 3D. For example, ranging circuitry for ToA or TDoA will not need much if any modification to work in 3D. The position computation involves mathematical calculation to simultaneously solve set of equations using node measurements. Therefore, the answer to first two components is: Yes. However, it is the localization algorithm which cannot be readily extended to 3D scenarios, hence creating the requirement for dedicated 3D localization schemes. Since localization algorithms usually also consider the physical constraints, anchor types, anchor movement and mobility, operational environment, etc., a 2D localization scheme may not necessarily be suitable for a 3D environment. Figure 1 depicts the scenario graphically.

The major contribution of this work is to present all the major 3D (generic, airborne, terrestrial, and submerged) localization schemes in a single literature. As opposed to most major surveys [3, 28, 30, 31, 60], this work is solely dedicated to 3D localization schemes. However, the most significant contribution of this work is to propose new classification based on functionalities of anchors for both generic and submerged networks and list their relative strengths and weaknesses.

Section 2 outlays the major anchor functionalities and requirements for different 3D localization schemes. Section 3 is arranged so as to provide brief introduction to essential localization taxonomy and explains the new classification. Section 3 also

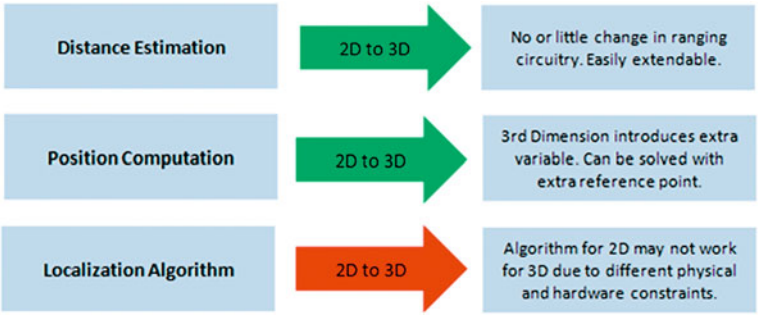


Fig. 1 Extensibility of 2D localization schemes to 3D. Distance estimation and position computation components of the 2D schemes can be extended to 3D. Localization Algorithm which involves practical implementation of the scheme, physical constraints, and hardware requirements cannot be readily extended to 3D

lists generic localization schemes/techniques according to their economic, messaging, delay, and computational cost. Section 4 lists major works in radio (generic) 3D localization schemes arranged according to the new classification. These schemes are either for airborne or terrestrial applications. Schemes which do not specify any application are also placed in this section. Section 5 is solely dedicated to localization schemes for submerged networks. Section 6 lists the salient features of each scheme along with its prospective applications. Finally, in Sect. 7 the open research problems in 3D localization are discussed.

2 Anchors Requirements in Localization

The practicality of a localization scheme essentially depends on the choice and role of employed anchors, their hardware requirements, deployment mechanism, mobility support, and density. To classify a localization mechanism based on practicality is prone to debate and counter arguments; since apart from anchors, localization delays, messaging overhead, convergence, accuracy, and resilience are also important metrics while determining the suitability of a localization scheme for a particular requirement. However, this classification weighs the practicality of anchor types to determine the relevance of overall localization scheme for a particular application, since a highly precise and resilient scheme employing highly complex and impractical anchors would be unusable for most prospective applications. In our analysis following factors are pivotal in determining the practicality of an anchor employed by a localization scheme.

2.1 Three-Dimensional Anchor Mobility

Some of the schemes require the anchors to be mobile in 3D plane. This might require anchors to be mounted on airborne independent platforms, e.g., drones, remotely piloted balloons, or autonomous submerged vehicles, etc. While airborne 3D mobile anchors [58, 77, 95, 97] require flight mechanisms, submerged 3D mobile anchors are usually retrofitted in autonomous underwater vehicles [26, 47]. However, most present schemes do not elaborate on mechanical fittings and design of such anchors which makes the practical implementation of such schemes a difficult realization. Furthermore, such anchors also require careful trajectory and mobility planning for complete localization coverage. Multiple anchors operating simultaneously would require sophisticated collaboration mechanisms. Furthermore, since such anchors usually have limited flight/submerged operational time, the localization scheme should not consist of long delays to avoid operational and maintenance costs of the anchor-mounting vehicles. Since economic cost of such platforms might be prohibitively high, careful investigation needs to be done to weigh benefits of using such a localization scheme for a prospective application.

2.2 Anchor Mobility in 1D

Interestingly, most submerged 3D localization schemes might require mobility of an anchor only in 1D plane [15, 25, 26, 47]. This might seem counterintuitive, but closer inspection reveals that a 1D mobile anchor can act as a good beacon source for the nodes deployed in 3D space. These anchors are also referred to as Detachable Elevator Transceivers (DETs) [25, 53, 104]. DETs usually operate by using a surface buoy as its launch platform. Once released by a surface buoy, DET descends along the depth of ocean while transmitting beacons. Though the schemes usually do not address the mechanical aspects of DETs, a practical implementation of a DET might face considerable constraints. DETs are prone to sway laterally due to ocean currents and swirls which might lead to transmission of erroneous beacons [22, 83]. The dive and rise mechanism of DETs would require rugged hardware. The cost of 1D mobile anchor depends on the type of mobility mechanism employed, which in turn dictates the practicality and relevance of the localization scheme for real-world scenarios.

2.3 Anchor Circuitry Requirements

A localization scheme works by using either range-based or range-free techniques. In either of ranging mechanisms, anchor plays a pivotal role acting as a beacon for the un-localized nodes. In both techniques, un-localized nodes receive or exchange beacons with the anchor nodes and estimate their positions. Depending on the scheme employed, estimation could be carried out centrally or in distributed manner. How-

ever, depending upon scheme and estimation mechanism employed, certain timing circuitry requirements and standards might need to be imposed not only on anchor nodes but also sensing nodes. For example, ToA-based schemes might require very tight synchronization requirements [27, 102]. TDoA-based schemes which operate using electromagnetic signals might require high resolution timing circuitry especially for radio-based ranging which increases the per node cost [43, 87, 93].

2.4 GPS Requirements

Accurate knowledge of self-position is paramount to the working of an anchor. Since the un-localized nodes in the network use positioning information of anchors as their reference, errors or inaccuracies in the anchor self-location estimation will propagate throughout the network, rendering the whole localization process erroneous. Therefore, it is essential that an anchor always has accurate knowledge of its position before it disperses its position information in localization beacons [7]. Apparently, the most logical methodology might be the use of GPS by anchor nodes to determine their own positions [97, 99]. Not surprisingly, this is also the most common approach in most localization schemes. However, for many schemes, depending on the environment, remote deployment fields, etc., the use of GPS might not be possible. For example, for many submerged 3D WSNs, submerged anchors [15, 102, 104] determine their positions using surface buoys. Some AUVs for submerged networks use sophisticated trajectory tracking mechanisms to keep track of their positions while beneath ocean surface. These surface buoys act as satellites for deeply submerged, remotely located, inaccessible anchors. Therefore, depending on application and environment, anchor self-localization can incur cost and may compose of complicated mechanical and software mechanisms. A tradeoff needs to be achieved between desired accuracy, system complexity, and economical cost of the anchor hardware.

2.5 Service Delay of Anchors

Most non-silent schemes require a complete handshake with the anchor node [26]. Based on request/response mechanism, the anchors send location information to nodes for position estimation. However, hardly any present work takes into account the service delays of anchors. An anchor serving a large number of sensing nodes might have long service (queuing) delays especially in acoustic systems. Since an anchor might have limited flight or submerged operational time (e.g., airborne anchors or AUV mounted anchors), long service delays might result in inhibitive economic cost for localization per node. Anchors might be required to carry out multiple sorties to fully localize a network resulting in huge operational and maintenance costs for the network.

2.6 Communication Range Constraints

Communication range can also be a significant factor in determining the relevance of a localization scheme for a specific application [52]. Communication range determines the influence region of an anchor. Increasing the communication radius might result in reduced anchor density for many silent schemes [16, 17]. However, for schemes which require two-way handshake between the un-localized nodes and anchors, increase in communication radius of anchor has to be complemented by increase in communication radius of sensing nodes also [28, 83]. This is not simple to achieve since it might require hardware and battery upgrades for sensing nodes. This might lead to preference for silent schemes. However, most silent schemes demand nodes to be synchronized or utilize complex ranging mechanisms for position estimation from anchor beacons. Resultantly, this increases computational, circuitry, and energy constraints for the sensing nodes.

2.7 Messaging Overhead and Hand shaking

Different localization schemes have different messaging overheads and hand shaking mechanisms between the anchor and un-localized nodes. Silent schemes do not exchange any messages with the anchors and just listens to the beacons transmitted by the anchors [16, 17, 27]. This gives the liberty of increasing the communication range of anchors since two-way communication between the nodes and anchors is not being realized. On the other hand, other schemes might require complete handshake between an anchor and a node. For example, TDoA-based schemes [28, 60] exchange messages with the anchors and uses time difference in arrival of message to estimate distance from anchors. Though such schemes might yield higher localization accuracy they require higher messaging overhead and close enough proximity of anchors to complete a handshake.

3 Essential Terminologies and Classification

In this section, we first briefly overview essential terminologies used in 3D WSNs followed by our proposed new classification.

3.1 Essential Terminologies

Localization algorithms may be classified based on several criteria, fundamental design, prospective application, network architecture, ranging mechanism, etc.

Table 1 Taxonomy and essential terminologies

Anchor based	Specially retrofitted nodes act as reference nodes. These nodes determine their absolute positions either directly by GPS (terrestrial or airborne networks) or indirectly by surface buoys (submerged networks) These anchors initialize the localization process by dispersing beacons in the network. Sensing nodes use these beacons to compute their absolute positions central or distributed manner
Anchor free	Coarse grained localization is achieved by using the relative positions of the nodes. These schemes usually employ connectivity information of the connected component of WSNs to determine the relative position of a node in a neighborhood
Range based	A method which employs radio or acoustic technologies to determine absolute distance/angle between two points for position estimation
Range free	Instead of using absolute distance/angle for position estimation, such schemes usually derive position estimate based on other metrics such as connectivity [14, 34, 44, 90], anchor proximity etc
Centralized	Nodes forward their range-based or range-free measurements to central sink. A sink can be fitted with computationally intensive and sophisticated position computation algorithm which produces globally optimized results. Once computed, coordinates are relayed back to the nodes
Distributed	Nodes compute their positions locally. Since measurements are not forwarded to the central sink, such schemes have lower messaging overhead and may achieve quicker localization. However, these schemes put higher computational load on nodes
Time of Arrival (ToA)	A ranging technology which employs signal propagation delay [9, 28, 62, 93, 96]. ToA circuitry may require tight synchronization and precision timing
Time Difference of Arrival (TDoA)	A ranging technology which uses time difference between arrivals of two signals (request/response mechanism). Many practical applications of this technology can be found [4, 9, 32, 33, 64, 70]
Angle-of-Arrival (AoA)	Nodes can use angle of arrival information of received beacons for position computation. By using AoA from multiple unique beacons, nodes can use triangulation to estimate their relative position to the anchors [56]
RSSI	The strength of received signal level is translated into distance estimation. It is essential that channel properties of the medium are known [5, 24, 28, 88]. RSSI based systems RADAR [4] and SpotOn [35] have been proposed for hardware constrained systems [33, 98]
Trilateration	By using the distance estimate from three reference points, equations for three spheres are formed and then solved simultaneously to yield intersection point. Accounting for ranging errors, unique solutions might not be possible [6]. Imitris et al. [50] explains the solutions to 3D trilateration in detail
Triangulation	If the bearings to reference nodes are known, simple trigonometry-based triangulation can be employed to determine relative position of the node [103]

Table 1 (continued)

Anchor-based	Specially retrofitted nodes act as reference nodes. These nodes determine their absolute positions either directly by GPS (terrestrial or airborne networks) or indirectly by surface buoys (submerged networks). These anchors initialize the localization process by dispersing beacons in the network. Sensing nodes use these beacons to compute their absolute positions central or distributed manner
Multilateration	This lateration method uses more than three reference points. This helps compensate for the ranging errors and achieve unique solution with greater accuracy
Bounding box	Computationally simple but less accurate method proposed in [79] which uses cubes instead of spheres for trilateration [6, 59]
Proximity-based	Usually centroid-based position computation method which takes simple averages of coordinates of neighborhood reference points to estimate its own position
Silent	Nodes do not exchange messaging with anchors or other reference points. Nodes achieve localization by listening to beacons
Active	Sensing nodes exchange messages with anchors and complete handshake
Synchronized	Nodes use globally synchronized high resolution timing circuitry and precision clocks
Non Synchronized	Nodes are not synchronized and use less expensive circuitry
DETs	Detachable Elevator Transceivers are special types of mechanically operated anchor nodes used in submerged networks which realize vertical motion
Surface anchors	Surface anchors usually remain on ocean surfaces and do not submerge. They use GPS to confirm their absolute positions and then act as satellites for submerged anchors
ToA versus TDoA	Shen et al. [76] comprehensively compares all the major range-based time-of-arrival (ToA) and time-difference-of-arrival (TDoA) location estimation methods for three-dimensional scenarios. It is recommended to refer to [13, 20, 76] for thorough comparison of ToA versus TDoA Analytical methods

Youssef et al. [96] succinctly present taxonomy for localization in WSNs which can act as a good serve for abridged introduction to generic taxonomy for localization in WSNs. Table 1 lists essential terminologies. Table 2 arranges different nonspecific generic schemes in descending order in terms of their economic, delay, messaging, and computational costs.

3.2 Classification Based on Anchor Functionalities

The 3-dimensional Wireless sensor networks can be broadly classified into two main categories: radio wave networks and acoustic networks. All infrastructure, terrestrial,

and airborne 3D networks employ radio waves for ranging and beaconing. Acoustic networks usually refer to submerged networks. Both categories of 3D networks have their specific design and implementation technicalities and have different requirements for the anchors. Based on the present scope of literature, a broad classification for both radio and acoustic networks is being proposed.

3.2.1 Radio Networks Classification and Naming Scheme

The study of literature reveals that localization schemes for radio networks can be broadly classified into four main categories depending on the anchor functionalities, position computation mechanism, complexity, and accuracy of the schemes.

The proposed naming scheme for the classification uses four letter abbreviations **XYZz**. The denotations are briefly explained below.

X: The single letter 'X' code refers to the mobility support in the anchor. 'S' stands for stationary anchors while 'M' stands for mobile anchors. Since mobility usually refers to anchors being mobile in 3D for radio networks, we do not specify the dimensionality of anchor mobility in the denotation.

Y: The single letter 'Y' code refers to mobility support in the sensing nodes of the network. 'S' stands for stationary sensing nodes while 'M' stands for mobile sensing nodes. Due to lack of literature on mobile sensing nodes in 3D WSNs, 'M' based classification is not included at present.

Zz: The two letter 'Zz' code represents the stand out feature of the localization scheme in terms of position computation mechanism. 'Ce' represents centroid based schemes; 'Cn' represents connectivity based schemes; 'Ln' represents lateration based schemes; and 'Ns' represents not specified. Most schemes employing stationary anchors fall in one of three classes: 'Ce' (centroid), 'Ln' (lateration) or 'Cn' connectivity based schemes. Some schemes employ mobile anchors with myriad of hybrid or new ranging and position computation mechanisms. These schemes are grouped under 'Ns' category.

SSCn (Stationary Anchors for Connectivity-Based Stationary Networks)

These schemes utilize connectivity information between the connected component of 3D WSNs to estimate node position. Beacons from stationary anchors are received by in-range nodes and then forwarded to distant nodes. Since the location information is derived from network connectivity, these schemes are not known for their accuracy especially for irregularly deployed networks. However, these schemes put least computational load on the nodes and are fairly unproblematic to implement.

SSCe (Stationary Anchors for Centroid-Based Stationary Networks)

These schemes have reduced complexity and practically simple to implement, and use range-free anchor proximity mechanism to achieve localization. Centroid-based

Table 2 Different schemes/technologies are arranged in descending order of respective sub-table heading

(a) Economic cost in terms of timing circuitry (descending order)	
Range based	Synchronized ToA-based schemes. Employ high resolution timers (high)
Range-based	Non-synchronized TDoA based schemes. Employ request/response mechanism (high cost but maybe less than ToA)
Range free	Non-synchronized. No special circuitry. Estimation using connectivity (acceptable cost)
(b) Messaging overhead	
Centralized	TDoA (very high)
Centralized	ToA (high)
Distributed	TDoA (high)
Centralized	Range free (acceptable to high)
Distributed	ToA (silent to low)
Distributed	Range free (low, merge with normal traffic)
(c) Localization delay	
TDoA	Centralized (very high, impractical)
ToA	Centralized (high)
TDoA	Distributed (acceptable to high depending on anchor positioning and range)
ToA	Distributed (low to high depending upon types of anchors)
Range free schemes (low to high depending upon scheme)	
(d) Computational cost/node	
Distributed schemes (acceptable to high)	
Centralized schemes (almost nonexistent)	
(e) Economic cost of anchors	
Fully mobile airborne (very high, special applications)	
Fully mobile submerged (very high)	
DETs (vertical motion submerged) (high)	
Surface buoys with GPS (affordable, numerous practical examples available)	

Schemes are arranged in descending order of (a) cost of timing circuitry, (b) messaging overhead, (c) localization delay, (d) computational cost per node, and (e) economic cost of anchors

determination is usually computationally simple but only yield coarse grained localization accuracy. These schemes are a reasonable tradeoff between design complication, computational and messaging overheads, algorithm complexity, and localization accuracy.

SSLn (Stationary Anchors for Lateration-Based Stationary Networks)

Lateration techniques (tri and multi-lateration) are fine grained (accuracy within 1m) ranging schemes which yield highly accurate results at the expense of increased computational and messaging overheads, algorithm complexity, and intricate ranging mechanisms. Depending on schemes, the anchors and sensing nodes might require to be retrofitted with high resolution timing circuitry, synchronous clocks, or sensitive transceivers. These schemes are relevant for sensitive applications which require high accuracy and can afford intricacies usually associated with such schemes.

MSNs (Mobile Anchors for Stationary Networks)

3D mobile anchors in radio schemes are generally envisioned to be retrofitted in drones or other airborne platforms, e.g., steered balloons. A mobile anchor for such schemes may have exorbitant hardware and operational costs limiting its use to applications sponsored by defense or resourceful corporations. These schemes can yield high accuracy but are prone to complexities arising due to ranging mechanisms, radio communications, trajectory planning, etc.

3.2.2 Acoustic Networks Classification and Naming Scheme

Acoustic networks are by far the most common 3D WSNs [84]. Hence the availability of literature for localization in submerged networks is far greater than radio based networks. This leads to six broad classifications for submerged acoustic 3D WSNs. The proposed naming scheme for the classification has **'XiiYi'** arrangement. We will briefly explain the denotations here.

Xii: 'Xii' represents the mobility support of anchor in the employed scheme. 'X' can be either 'M' (meaning mobile) or 'N' (meaning not-mobile). The two lower case letters 'ii' only appear for 'M' (mobile) scenarios and essentially describe the degree of anchor mobility in the network. The 'ii' can be either '1d' or '3d', representing motion of anchor in 1 dimension and 3 dimensions respectively. The 'ii' does not appear when preceded by 'N'.

Yi: 'Yi' represents the dominant trait/feature of the scheme which dictates the position computation mechanism employed. 'Y' can be either 'C', which represents centralized schemes; or 'D', which represents distributed schemes. The lower case trailing 'i' denotes the salient feature of the centralized/distributed scheme. The 's' indicates synchronous schemes; 'a' indicates synchronous free schemes; 'v' indicates localization scheme extendable for moving nodes such as AUVs; and a missing 'i' denotes 'un-specified'.

M1dC (Vertical Motion (1D Mobile) Anchors for Centralized Schemes)

These schemes employ vertically mobile anchors which accomplish motion in one dimension for submerged acoustic 3D WSNs. Since the position estimation for such

schemes is done centrally, these schemes are 'non-silent' and may incur considerable extra network traffic, long acoustic CSMA delays and resultantly long localization delays. However since sink can be retrofitted with computationally intensive and sophisticated position estimation mechanism, these schemes can yield high accuracy and relieves computational load on nodes due to centralized estimation.

M1dDs (Vertical Motion (1D Mobile) Anchors for Synchronized Distributed Schemes)

These schemes use anchors which also realize motion vertically (1D). However their distributed synchronous nature allows them to be implemented as 'silent' (for ToA-based systems) and also eliminates long sink service and communication delays of typical acoustic networks. However, their distributed nature puts greater computational load on the nodes.

M1dDa (Vertical Motion (1D Mobile) Anchors for Non-Synchronized Distributed Schemes)

Flexible in design, these schemes offer numerous different designs based on ranging mechanisms and anchor-node handshake. Their non-synchronized nature allows elimination of the need for maintaining tight synchronization globally which reduces hardware and circuitry complexity as well as economic costs.

ND (Non-Mobile Anchors (DET Free) Distributed Schemes)

These schemes do not employ conventional 1D or 3D mobile anchors; rather anchor nodes are similar in mechanical abilities to other sensing nodes in the network. Stationary anchors initialize the localization process. These schemes can be either range or connectivity based.

M3dD (AUV (3D Mobile) Anchors for Distributed Schemes)

A 3D mobile AUV for submerged operations can be retrofitted to acts as anchor and transmit beacons underwater to initiate localization process for the sensing nodes. AUV is usually a very expensive hardware with sophisticated on board piloting systems for autonomous maneuvering. AUV trajectories also need to be carefully pre-planned to provide complete localization coverage. Though use of AUV brings flexibility in the localization scheme and may also help improve accuracy, the hardware, and operational costs for 3D mobile anchors restrains their use to very limited submerged applications.

NDv (Fixed Anchors for AUV Localization)

Though most AUVs have onboard sophisticated trajectory tracking mechanisms to reliably determine their own positions; it is possible that AUVs reconfirm their positions from fixed anchors while submerged for improved accuracy. Some of the present schemes for 3D localization can be easily evolved to be used for these purposes. Apart from providing localization to conventional stationary sensing nodes, these localization schemes can have applications in navigational assistance, collision avoidance, and steerage through obstacle-prone submerged environments for AUVs.

4 Three-Dimensional Localization Algorithms

The classification of localization schemes is a moot point and open to debate. Different authors have classified the localization algorithms in numerous different ways. For example, Guangjie et al. [31] classify the algorithms based on mobility of sensor nodes and reference nodes. Resultantly there are four main classes in [31]: static nodes, static anchors; mobile nodes static anchors; static nodes, mobile anchors; and mobile nodes, mobile anchors. Some authors classify based on centralized and distributed algorithms. Range-based classification is also commonly used for localization schemes. It is fairly common that schemes are further sub-classified based on centralized/distributed, mobility, active/silence, etc. However, due to non-presence of vast literature in 3D localization in WSNs, such extensive classifications are not relevant for 3D WSNs at this moment. Instead, a new classification is being proposed based on anchor mobility and position computation method employed by the generic radio-based 3D WSNs. The new classification not only covers the present literature in this field but also classifies schemes into groups based on complexity and accuracy.

Note: Some of the abbreviations for the localization schemes used in the section might have been altered from the original work to avoid conflicting abbreviations for different schemes.

4.1 SSCn (Connectivity-Based Stationary Anchors)

In terms of complexity and hardware requirements of anchors, schemes which utilize connectivity based stationary anchors are inherently similar to range-free centroid-based stationary anchors. The fundamental difference is that the beacons are essentially embedded into the network traffic. Although this might sound appealing from the perspective of reduced messaging overhead, however, most such mechanisms require sensing nodes to exchange and forward messages. At the network level, this actually results in increased messaging overhead per node. Furthermore, connectivity-based schemes are also trickier to implement and require careful and considerate planning. These schemes also tend to have prohibitively long localiza-

tion delays. Wang et al. [46] propose a Distance Vector DV-Hop algorithm for 3D space. Nodes receive the beacons from anchors, increment the hop count and forward to further nodes. Beacons also monitor the traffic exchange and internally estimate average per hop distance in the network. The un-localized nodes extract this information from beacons to estimate their own position. Though the scheme is fairly simple and computationally less intensive, if not implemented vigilantly, it can result in exorbitant increase in messaging overhead as well as localization delays. Furthermore, since hop distance is used for location estimation, some localization error is inherent in the scheme. Shu et al. [78] also propose a DV-based localization algorithm. The algorithm performs in three phases: compute the least hop counts, translate into distance between node and beacon node, and determine node position using Assumption Based Coordinate (ABC) [69] algorithm. The main feature of the algorithm is its simplicity. However, localization error of as high as 40 % might result for irregularly placed nodes in the network. Zhao et al. [100] outlay an important prospective application for connectivity-based localization for 3D Surface localization. In 3D surfaces, the wireless signals only propagate along the surfaces with no direct correlation between Euclidean distance and shortest path between nodes. For example, sensors deployed on the surface of mountain, uneven platforms, etc. Easily available and economical on-board pressure sensors measure the height in each node. The problem is then treated in 2D with connectivity information of the graph being projected onto x–y plane. The projected triangular mesh is used to yield localization solution. Since the variables are reduced due to use of pressure sensors, the scheme yields high accuracy. The authors have also practically implemented the scheme with good results. The scheme offers a practical design for many complex 3D surfaces. However, it is to be noted that the algorithm works for single value complex surfaces and may fail to yield solution for more intricate 3D surfaces. Shang et al. [74] also use connectivity to estimate distance when there are no anchors with absolute position. Using node connectivity, it builds a distance matrix. It then uses this distance matrix to layout a map with relative positioning of nodes. The scheme is computationally intensive in order of $O(x^3)$ but offers great flexibility and scalability in design. Numerous flexible variants of the scheme can be implemented. For applications which may not have reliable anchors and can afford slight computation overhead, this relative positioning mechanism can be a reasonable resort. In conclusion, it can be said that connectivity-based schemes which employ stationary anchors usually are simple to implement and practical. They do not have specific hardware and mobility requirements for the anchors. However, these schemes not only have high messaging overhead and long localization delays, their location estimation is also not highly accurate due to fact that nodes infer their position by using average hop distance in the network. However, these schemes can be a reasonable choice for small evenly deployed 3D networks with uniform hop distance. Refer to [55] for comprehensive analysis of DV schemes (the work is for 2D systems).

Advantages: These schemes have simple and flexible design and put less computational load on the nodes. They have Low implementation cost and do not need sophisticated on-board timing circuitry for the nodes. Vigilant implementation reduces messaging overheads.

Disadvantages: These schemes have low localization accuracy and are also prone to propagating errors throughout the network. Schemes might incur long localization delays and messaging overheads. Accuracy greatly reduces for irregularly deployed networks.

4.2 SSCe (Centroid-Based Stationary Anchors)

Schemes based on Centroid algorithm are practically simple and can be implemented as silent if desired. These schemes are anchor-proximity schemes which use range-free mechanism to achieve localization. Anchors broadcast their position information to all nodes within their transmission range. Nodes listen to these reference beacons and estimate their position using centroid determination (usually averaging of coordinates of anchors). Though centroid algorithms are computationally simple and easy to implement, they mostly only achieve coarse-grained localization and are prone to considerable inaccuracies. Chen et al. [14] build on the work presented in [90] and use centroid theorem of coordinate-tetrahedron in the volume coordinate system to estimate the localization information for 3D WSNs. By randomly selecting four anchors and forming a virtual tetrahedron, it uses centroid theorem to calculate barycenter of each tetrahedron. Averages of many barycenters yields estimated position. Although this tetrahedron-based centroid algorithm approach increases computational load on the nodes, it helps alleviates considerable estimation inaccuracies compared to typical centroid algorithms. Bulusu et al. [8] show that tetrahedron localization algorithm can improve accuracy by 29 % compared to traditional algorithms but at the expense of increased computation. Furthermore, it is noteworthy that since a two-way handshake is not completed in this scheme, anchors can transmit over long ranges hence minimizing the requirements for number of anchors required. This also enables silent implementation of the scheme. However, compared to typical centroid algorithms, tetrahedron based centroid approach requires at least four anchors to achieve localization. Lai et al. [44] also use centroid algorithm which makes use of geometric relationships and communication constraints between the nodes. Instead of using a 3D polyhedron such as a tetrahedron, [44] constructs a 3D graph and then maps it to a 2D plane. It lays out an analysis mechanism to estimate node position. Interestingly, localization ratio approaches as high as 99 %—a very high ratio for a centroid-based algorithm—when the anchor (reference) node density is greater than 6. However, this scheme can result in considerable high computational load. The computational overheads are coupled with messaging overheads when this scheme is implemented in distributed manner. Lv et al. [49] present a spherical shell overlap algorithm. Although this scheme does not follow the complete definition of a centroid algorithm, it has intrinsic similarities with centroid-based systems. Concentric spheres are used to virtually divide the space with anchors being visualized at the center of these spheres. By considering the spherical centers of the adjacent spheres and spherical overlap information, the unknown node coarsely estimates its position.

The scheme requires extremely low computation and messaging overhead but only offers low localization accuracy.

To abridge, centroid-based localization algorithms are a good choice for coarse-grained range-free location estimation. Though their results are prone to inherent error, such schemes can be a sensible choice for many low-end applications. Since these schemes usually do not require extra hardware, mobility mechanism or high resolution timing circuitry for anchors; anchors in such schemes can be deployed as stationary reference points. The reduced complexity in anchors usually offsets the slight increase in cost due to higher number of minimum anchor required in such schemes.

Advantages: Schemes offer reasonable accuracy (higher than connectivity-based schemes) and comparatively low computational load. They do not require sophisticated on-board timing circuitry for the nodes and also offer flexible design.

Disadvantages: Not suitable for applications demanding very high localization accuracy. Nodes need to be in range of at least four anchors/reference nodes.

4.3 SSLn (Lateration-Based Stationary Anchors)

Lateration is a rangebased technique that uses distance measurements between anchor and sensing nodes. While the tri-lateration uses three reference points, multi-lateration generally uses 4 or more reference points for 3D space [21, 29, 43, 72, 73]. This makes most multi-lateration techniques a fine grained (accuracy within 1m) ranging schemes which yield highly accurate results, however, at the expense of usually increased computational and messaging overheads. Kuruoglu et al. [43] propose a multi-lateration based scheme (3D-AML) which uses concept of intersecting spheres to estimate distances. The intersection of two spheres yields equation of circle. The fourth reference point is then used to determine the final position without ambiguity. Though the scheme is considerably accurate for most applications, it also has higher computational cost per node. Furthermore, since the model assumes transmission ranges of anchors to be uniformly omnidirectional, this can make the model inaccurate in obstacle prone environments. Detailed topographic maps may need to be required to have accurate modeling of communication ranges of anchors. Inaccurate modeling may reduce the accuracy to conventional centroid-based schemes while the computational overhead would still be high. Doukhnitch et al. [23] tend to reduce some of the extra computational load incurred in lateration-based schemes by using a trilateration based method (DODS) which uses vector notations and simple add and shift operations. The vectors are recursively rotated towards each other to find intersection points. This method is named Dynamic rotation method with fixed angle and is based on work in [68]. Refer to [23] and [68] for detailed explanation and graphical depictions. The resultant method achieves a good compromise between accuracy, complexity, and computational overheads. It is claimed in [23] that computational load could be relieved as much as 67 % by employing DODS. The only drawback is the dependence on recursive steps for solution. This can be

a potential pitfall since this may lead to error magnification in some scenarios. If a prospective application requires considerable accuracy and can afford extra computation inherent with lateration-based methods, Manolakis uses trilateration in [50] which compensates for reasonable errors in range measurements by incorporating polynomial-type form of algorithm. Similarly, Thomas et al. [87] use constructive geometric arguments for trilateration, and derive a formula containing a few number of Cayley-Menger determinants which allows easier identification of erroneous results. Refer to [87] for more details. Zheng et al. [101] propose an interesting localization algorithm which is inspired by the living habits of bees in physical world. Though the method is not specifically based on lateration, it uses received signal strength, cosine laws and mobility models of bees to determine relative positions. Amusingly, the scheme yields high accuracy, although the hardware and computational constraints make the practical realization of this algorithm infeasible for most applications.

In conclusion, lateration-based techniques which employ stationary anchors yield very high accuracy at the expense of increased computational load for the nodes. These techniques are mostly range based and use distance information from at least 4 anchors in 3D space to compute the position. However, this accuracy is dependent on rigorous calibration of ranging mechanisms such as RSSI, ToA, etc. Some ranging mechanisms call for special hardware and high resolution timing circuitry increasing the economic cost. Furthermore, the number and optimal positioning of anchor nodes for maximal ranging coverage can also be a potential design limitation.

Advantages: High localization accuracy.

Disadvantages: High computation load for nodes. Nodes require sophisticated ranging mechanisms (timing circuitry, signal detection). Complex lateration-based position estimation techniques need to be efficiently applied on resource constrained sensing nodes. Erroneous beacons introduce error which might offset the increased accuracy of lateration-based schemes.

4.4 MSNs (Mobile Anchors for Stationary Networks)

Mobile anchors can be used as beacon sources for localization in 3D networks. Not surprisingly, this is neither simple nor cost effective for most applications. Ideally, mobile anchors fly over or travel through the network using their mobility mechanism and transmit beacons. Nodes use this information to estimate their own positions. Using mobile beaconing can be cumbersome task since a lot of parameters such as anchor speed, beacon transmission frequency, transmission range, etc., need to be carefully configured. Furthermore, mobility patterns also need to be preplanned. However, the biggest obstacle to using mobile anchors in most schemes is its intimidating economic costs. Refer to [2] for a detailed survey on mobile wireless sensor localization. An autonomous mobile anchor needs a propulsion mechanism, resilient communication systems, safety mechanisms, extended battery backups, take off (launch, initialization), and landing (finish, end) systems along with vigilant

maintenance. These all factors when compounded together can result in significantly high costs for the localization scheme. Therefore mobile anchors are usually selected for applications which have very specific requirements and their operational environments do not allow for stationary anchors. For example, in defense applications, airborne mobile anchors could be used for localizing nodes on the ground. For nodes installed in extremely remote or hostile environments, mobile anchors could help initialize the localization procedure. Yadav et al. [95] outlay a range-free localization technique which uses GPS-equipped mobile anchors as beacon sources. Static nodes listen to beacons and update their entries. Using the connectivity information with mobile beacons, nodes estimate their positions. Authors claim that this approach is less computationally intensive and requires 98 computations (54 multiplications and 44 additions) as opposed to 153 computations (88 multiplications and 65 additions) in [58, 95]. Furthermore, this scheme allows the node to compute absolute position since GPS information is embedded in beacons. Also, since a same anchor can transmit from multiple locations, the un-localized nodes have multiple unique beacons for reference which results in high accuracy. Ou and Ssu [58] also use flying anchors, which fly through the sensing space transmitting their current locations. The basis of this scheme is the geometric corollary that states that a perpendicular line passing through the center of sphere's circular cross section also passes through the center of that sphere [58]. The sensor nodes listen to the beacons and maintain a visitor list. After receiving four unique beacons, circular cross sections are constructed to determine position. Authors highlight that with transmission radius of 15 m, spherical cross section based approach has localization error of 1.6 m compared to 2.4 m for centroid algorithms. However, it also needs to be noted that transmission radius of just 15 m would require the anchors to fly exceptionally low which might require additional expensive low-flying collision avoidance systems. Yu et al. [97] also propose the use of airborne mobile anchors and has same system setup as [58, 95]. However, in [97] the sensor nodes select the set of best RSSI beacon updates and use them to construct non-parallel planes perpendicular to anchor trajectories. The scheme puts almost negligible computational load on sensing nodes. However, RSSI-based measurements are prone to error unless accurate channel models are available. Zhang et al. [99] address the problem of 3D surface localization by using air borne mobile anchors as functional dual of target tracking. Terrestrial nodes track the moving targets by listening to beacons and inferring their own positions. The use of mobile anchors for 3D surface and using target tracking allows the localization scheme to work independently of node densities and hence robust to complex 3D environments.

It is interesting to note that most airborne mobile anchor-based schemes silent in nature and do not require a handshake between an anchor and the sensing node. Sensing nodes can listen to beacons transmitted by distant airborne anchors and estimate their own positions. Furthermore, since an anchor can transmit beacons from multiple locations, a single airborne anchor might be enough to help achieve localization for a reasonably sized network. As opposed to other coarse grained range-free schemes, mobile anchor based schemes (especially airborne mobile anchors) have reasonable high accuracy. These characteristics make such schemes a fitting

Table 3 Salient features of range-free schemes

Range free scheme	Design hazards	Advantages
Anchor proximity (centroid) [8, 14, 44, 49, 90]	Optimum anchor positioning. Accuracy dependent on number of anchors. Nodes need to be within anchor range	Low Messaging overhead. Can be implemented as silent schemes. Small localization delay. Prone to high localization errors
Connectivity [46, 78, 95]	High messaging overhead. Prone to long localization delays. Inherent estimation error	Extremely simple to implement. Can be incorporated with normal traffic. Resilient

choice for many demanding applications, e.g., defense, hazardous area monitoring, etc. However, the usually excessive cost of airborne mobile anchors might deter their use in most low-end practical applications of 3D WSNs.
Advantages: High localization accuracy can be achieved. Trajectory planning allows for comprehensive localization coverage. Single an anchor can transmit multiple unique beacons from different locations; this eliminates the need for multiple anchors.
Disadvantages: Exorbitant hardware and operational costs for anchors. Anchors require careful trajectory planning and flight paths. Extended operations for the anchors might not be feasible.

4.5 Additional Remarks on Range-Free and Range-Based 3D Localization Schemes

Range-free localization schemes are relevant for applications which do not have stringent localization accuracy requirements (error of 20–40 % is common [78]) and require computationally simple algorithms. These schemes usually estimate position either by connectivity information or anchor proximity. While connectivity-based range-free schemes are also prone to high messaging overhead (DV-Hop, DHA), network flooding and long localization delays; anchor proximity (centroid)-based schemes suffer from anchor positioning issues. However, intelligent and efficient implementation can offset some of these bottlenecks. For example, the messaging overhead in DV-Hop can be reduced by incorporating localization information in the normal network traffic (refer Table 3).

Generally speaking, range-based schemes usually have high localization accuracy conditional to rigorous calibration of ranging mechanisms. Some ranging mechanisms call for special hardware and high resolution timing circuitry increasing the economic cost. Furthermore, the number and optimal positioning of anchor nodes for maximal ranging coverage can also be a potential design limitation. Table 4 lists salient features of range-based schemes covered in this section.

Table 4 Salient features of range-based schemes

Range-based type	Design hazards	Advantages
RSSI based [97, 99, 101]	Rigorous calibration of RSSI based estimation hardware required. Inherent error in estimation	Hardware less expensive compared to ToA, TDoA. Low computation overhead. Schemes can be implemented as silent schemes
ToA, TDoA [43, 68, 87, 93]	Complex timing circuitry. Synchronization might be required in ToA. Expensive hardware. Computation for signal processing	Very high accuracy especially in submerged environments. RF based ToA schemes can have very long ranges

5 Underwater Acoustic Sensor Networks (UASNs)

UASNs are by far the most common practical three-dimensional Sensor Networks with increasingly important applications in oceanographic data collection, ocean sampling, environment and pollution monitoring, offshore exploration, disaster prevention, tsunami and seaquake warning, assisted navigation, distributed tactical surveillance, search and rescue, and mine reconnaissance [63, 89]. Unlike many terrestrial sensor networks which can be modeled as two dimensional networks, the sheer large size and scope of UASN deployment fields and height variations in UASN deployed nodes make it essential to visualize such networks as 3D in nature.

Acoustic communications is the typical physical layer technology for most underwater sensor networks. UASN poses unique challenges due to harsh underwater environment, such as limited bandwidth capacity [80], high and variable propagation delays [65], high bit error rates, and temporary losses of connectivity caused by multipath and fading phenomena [81]. Kantarci et al. [28] presents a comprehensive survey for localization techniques for UASNs. Although the work is not specific to 3D networks it adequately covers the major works carried out in this field. Partan et al. [60] discusses major challenges and practical issues for underwater sensor networks. The survey work in [82] and [91] discusses most of UASN localization schemes in detail. In the following text we briefly study the relevance of different ranging technologies for UASNs.

ToA. It is much preferred choice in UASN than in terrestrial systems. Since sound travels at fraction of speed of light, the level of sophistication required for ToA-based ranging of acoustic signals is less and also more economical.

RSSI. Time-varying properties of the ocean environments make RSSI based ranging mechanisms unreliable [42]. It is not a preferred choice.

AoA. Though used in numerous schemes, it requires use of directional acoustic transceivers and careful consideration for anchor positioning and directional beacons [94].

TDoA. Able to achieve localization without global synchronization, TDoA yields good accuracy and high reliability. There are several schemes employing TDoA for UASNs [28].

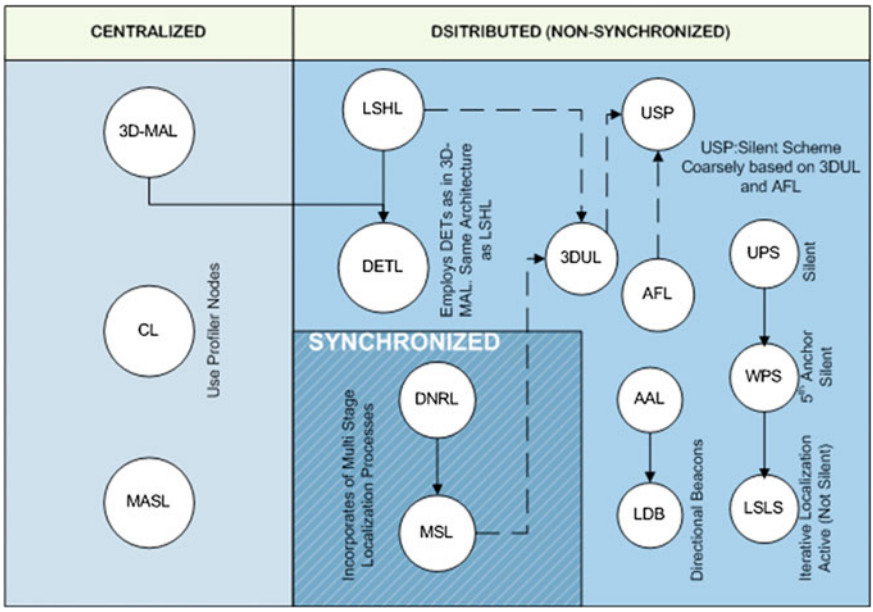


Fig. 2 Evolutionary arrangement of the submerged 3D localization schemes. Arrangement is also based on centralized and distributed localization; synchronized and non-synchronized operations. The direction of *arrow head* shows an evolved scheme. *Broken line* indicates that scheme is coarsely based on previous scheme

Figure 2 depicts the major schemes covered in this section. It also indicates the evolutionary relationship between different schemes and salient standout features/characteristics of evolved schemes.

As opposed to most other classifications, the proposed new classification is based on two main metrics: degree of anchor mobility and centralized/distributed nature of localization scheme. This classification not only directly translates into practicality of a scheme; but also indicates complexity, accuracy, and economic cost of the 3D localization schemes.

5.1 M1dC (Vertical Motion (1D Mobile) Anchors for Centralized Schemes)

Vertically mobile anchors accomplish motion in 1 dimension for submerged acoustic 3D WSNs. Such anchors can be referred as Detachable Elevator Transceivers (DETs) [28, 104]. They use surface buoys to change their depth. Anchor can be mechanically tied to the surface buoy and can change its depth either by buoyancy control or cable-wind/unwind operations. Since DETs are designed to realize only vertical

motion, their absolute coordinates for x–y plane remain the same as their respective surface buoys. Surface buoys get their location using GPS and relay it to DETs. DETs can determine their own depth by employing economical submerged pressure sensors which have considerable accuracy. Zhou et al. [104] use DETs (scheme called 3D-MALS) which descend and transmit beacons at varying power levels along the way. The un-localized nodes send their lowest power reading to the sink. Based on the lowest power readings of a node from different anchors (or different beacon series), the sink estimates the position and conveys it back to node. Ideally, the scheme should yield high accuracy. However, due to ocean currents, DETs may drift laterally and transmit erroneous beacons. This error can be compensated by incorporating sophisticated ocean current modeling in the centralized decision maker (sink). Mirza and Schurgers [53] use buoyancy control in anchors (referred to as drogues in text). However, to account for lateral drift due to ocean currents and reduce error in absolute position estimation, use of profiler nodes is also proposed. Drogues track the trajectories of profiler nodes which act as moving references. The scheme reduces error in estimation but at the expense of increased design and implementation complexities. In practical scenarios, realization of such multi-stage localization scheme with multifarious hardware would not be an easy accomplishment. Therefore, although these schemes have high accuracy and relieves computational load on nodes due to centralized estimation, they are prone to significant errors if ocean currents modeling is inadequate. Furthermore, centralized estimation comes at the price of 'silent-localization.' Since nodes have to send their measurements to sink, considerable extra network traffic could be incurred. For acoustic networks this might result in exorbitant long delays in terms of sink service availability and acoustic CSMA. For example, it is explained in [54] that to limit collisions to less than 5 %, acoustic CSMA back off may be as large as 300 s for a 10 byte packet to be transmitted over a distance of 500 m. However, centralized processing allows for globally optimized results since sinks not only have network wide positioning estimates, but can also be equipped with sophisticated ocean current modeling system to compensate for undesired lateral movements of anchors. These schemes can be an appropriate choice for delay tolerant submerged applications with computationally limited nodes.

Advantages: Globally optimized position estimation can be done at the central sink. No computational load on the nodes.

Disadvantages: Nodes need to forward their measurements to central sink. Long sink service delays might be incurred. High messaging overhead compared to distributed schemes.

5.2 M1dDs (Vertical Motion (1D Mobile) Anchors for Synchronized Distributed Schemes)

Vertically mobile anchors (1D Mobility) for distributed schemes usually have same mechanical requirements for anchors as centralized schemes except that nodes compute their own positions instead of forwarding their measurements to central sink. This allows for the schemes to be implemented as 'silent' (for ToA-based systems) and also eliminates long sink service and communication delays of typical acoustic networks. Erol et al. [27] proposes a synchronized distributed scheme (DNRL) which uses buoyancy controlled anchors similar to [53, 104]. The DETs (anchors) dive and rise periodically while broadcasting beacons. Nodes listen to the beacons and use ToA based measurements to determine their own positions. The major drawback of such schemes is their synchronization-based distance estimation. All nodes need to be fitted with synchronized high resolution accurate clocks. A synchronization offset is difficult to detect and will lead to fallacies in estimation. However, since localization can be achieved without a handshake between mobile beacons and nodes, long range beacon signals can be used to localize nodes located in remote hostile environments. These schemes can be a good choice for sparsely dense, remotely located networks with computationally capable nodes. Another limitation of such schemes is the requirement for each node to be within range of at least 4 unique beacons. The requirement relaxes to 3, if nodes use pressure sensors to estimate their depth and anchor beacons to determine their positions in x-y plane. However, in any case, this can be bottleneck for beacon-out-of-range nodes. Erol et al. [25] address this problem by introducing a supplementary phase in localization process. The scheme is called MSL [25] and is an evolution of DNRL [27]. The newly localized nodes in MSL act as coarse beacons sources for beacon-out-of-range un-localized nodes. These coarse-localized nodes can recalculate their position once they receive DNRL beacon signals. The result is not only greater localization ratio for the network but also at quicker pace. Kantarci et al. [28] analyze the MSL and explain that it is one of few localization schemes for acoustic networks which utilize realistic underwater mobility model. However, by forcing localized nodes to transmit coarse beacons (coarse since these beacons are prone to compound error) MSL losses its 'silent' trait. But it is not a dreadful tradeoff since these beacons can be embedded in network traffic and extra messaging overheads could be limited to safe margins.

Advantages: Synchronized ranging mechanisms yield considerable accuracy.

Disadvantages: Tight global synchronization requires expensive high resolution timing circuitry and precision clocks. Synchronization offsets are difficult to detect.

5.3 M1dDa (Vertical Motion (1D Mobile) Anchors for Non-Synchronized Distributed Schemes)

Non-Synchronization-based distributed schemes offer flexible design. Not only such schemes relieve the node to maintain tight synchronization globally but also put less messaging overhead on loads due to distributed localization processing. Chen et al. [15] propose a scheme referred as DETL, and employs Detachable Elevator Transceiver. The scheme is an improved variant of LSHL [102] (covered under non-mobile surface buoys as anchors). DETs realize motion in one dimension (vertically) and transmit beacons for the network nodes. The network nodes contain a small percentage of high power nodes (secondary anchors) which use DET beacons for self-localization. These anchors then initiate localization process for ordinary nodes in the network. DETL tends to mitigate some of the deficiencies in LSHL. For example, [15] eliminates the need of long range communication requirements between anchors and surface buoys in LSHL at the expense of additional DET hardware. Although the scheme does not specifically outlay itself as synchronization-free, its simple bounding box-based position estimation method allows for flexible implementation.
Advantages: Non-synchronized schemes are easier to implement, more flexible, and require less expensive timing circuitry for acoustic networks.
Disadvantages: Prone to longer localization delays. Some schemes require handshake with the anchors which incurs messaging overheads and makes them 'non-silent.'

5.4 ND (Non-Mobile Anchors (DET Free) Distributed Schemes)

If an anchor refers to mechanically operated DET in submerged networks, these schemes can then also be classified as anchor-free schemes. These schemes do not employ conventional 1D or 3D mobile anchors; rather anchor nodes are similar in mechanical abilities to other sensing nodes in the network. These schemes can be either range based [102] or connectivity based [57]. For range based, these anchors (reference nodes) can estimate their position using their proximity to surface buoys and then act as beacon sources for other distant nodes in the network. Connectivity-based schemes have lesser accuracy and do not employ any anchors or surface buoys, and just use connectivity information of the network to estimate the relative positions of the nodes. These schemes are economical and comparatively easier to implement. Such schemes usually relieve intricate mechanical hardware requirements of submerged anchors at the expense of accuracy in localization process. Othman et al. [57] propose an anchor-free scheme (AFL) which requires no submerged anchors. A seed node initiates the node discovery process. As the network discovery proceeds, nodes maintain connectivity map of their neighborhood and use it to estimate their relative position in the network. As expected, the process incurs considerable messaging overhead. However, if implemented carefully and efficiently, the messag-

ing overheads can be mitigated to acceptable levels. Furthermore, during deployment, vigilant recording of sensor IDs and their deployed field region has to be done. This allows for mapping relative position of nodes to their actual positions in the deployed 3D field during localization process. The scheme has low localization accuracy, prone to long delays and yields only relative position of the nodes based on their connectivity matrices. However, it has relevant applications for many scenarios where mechanically operated anchors and surface buoys are not readily available. Zhou et al. [102] propose a range-based hierarchical scheme called LSHL which do not employ mobile anchors. Submerged anchors (similar in mechanical abilities as other nodes in the network) employ their proximity to the surface buoys and use range-based mechanism to determine their position. Using one-way ToA ranging and lateration, ordinary nodes listen to beacons from submerged anchors and determine their positions. Once localized, a node starts acting as a reference node for the other distant nodes in the network. Though the scheme is economical to implement, one-way ToA requires tight synchronization. Furthermore scheme is also prone to compound error in location estimation for distant nodes in the network. However, the economic and implementation practicality of scheme makes it a suitable candidate for many applications which do not have stringent accuracy requirements. Isik and Akan [40] also use just three surface buoys to initialize localization in a scheme referred as 3DUL. Similar to [102], localized nodes assume roles of beacon sources for other nodes in the network. However, as opposed to ranging mechanism in [102], 3DUL uses two way ranging mechanism and lateration to estimate node positions. This relieves the nodes from tight synchronization requirements but requires two-way handshake between reference nodes and un-localized nodes. This can incur significant messaging and delay overheads due to acoustic CSMA. The scheme is also prone to magnifying and propagating errors in its iterative cycles. However, this scheme should perform well for sparsely dense networks. Teymorian et al. [18, 85, 86] propose a localization framework termed USP. USP allows 2D terrestrial ranging mechanism to be extended for 3D networks. By employing non-synchronization-based ranging mechanism and a location projection technique that maps the position of reference nodes onto a same plane, USP is able to achieve localization for the sensing nodes. Zhou et al. [22] propose a hierarchical scheme (SLMP) which uses a submerged anchor nodes with long range communication with surface buoys to determine their positions. Sensing nodes use anchors to localize themselves and can also assume the role of reference nodes to increase localization coverage in the network. The major contribution of the scheme is incorporation of node movement due to water currents and dispersion. It is explained in [22] that movement of underwater objects is dependent on many environmental factors including current and temperature. Furthermore, mobility characteristics of submerged objects vary markedly in different environments and ocean depths. For example, submerged nodes near sea shore demonstrate semi-periodic property because of tides. The work in [22] investigates mobility characteristics in shallow seashore areas and effect of tidal currents on node movements. Nodes predict their future mobility patterns according to their past position information and estimate their present location. Anchor nodes continuously check for error offset between

predicted location and measured location. If the error is within the stipulated limits, the mobility model is assumed valid. Otherwise anchor re-executes mobility prediction algorithm and re-estimates its position, followed by regeneration of beacon updates for sensing nodes. The scheme is one of few works which incorporates node mobility due to ocean currents in its fundamental model. The scheme is especially relevant for seashore based 3D WSNs deployments. The work in [22] can be extended or incorporated for other localization schemes to improve localization accuracy.

Advantages: Mechanical complexities of mobile anchors are relieved. Schemes are easier and more flexible to implement. Vast presence of literature allows for myriad of selection options.

Disadvantages: Higher anchor densities with absolute coordinates might be required compared to vertically mobile anchors.

5.5 M3dD (AUV (3D Mobile) Anchors for Distributed Schemes)

AUVs are autonomous underwater vehicles with propulsion system allowing them to move in 3 Dimension. An AUV is usually a very expensive hardware with sophisticated on-board piloting systems for autonomous maneuvering. An AUV can get its GPS coordinates while on surface and can keep track of its position while submerged by carefully tracking its trajectory underwater. AUV acts as anchor and transmit beacons underwater to initiate localization process for the sensing nodes. Therefore, AUV based schemes are relevant for very critical applications with huge budgets, e.g., defense or underwater oil exploration, etc. Erol et al. [26] propose AUV-based scheme (AAL). The scheme is intuitively simple. It uses three basic messages: wakeup, request and response. AUV sends a wakeup message. The nodes respond with a location request message. AUV responds with its location update. Round Trip Delay of request/response exchange with AUV is used to determine position of nodes. Multiple (at least three) exchanges with AUV are used in triangulation by the nodes to determine their positions in x–y plane and pressure sensors are used to determine the depth. Since AUV trajectories can be pre-planned to cover remote inaccessible niches, the design brings flexibility in the system complemented by reasonable accuracy. Furthermore, lack of synchronization requirements also allow for less expensive on-board timing circuitry for sensing nodes. However, since the scheme is fundamentally based on multiple exchanges with AUV, acoustic CSMA could be a considerable bottleneck. Furthermore, as pointed out in [28], the accuracy of [26] may also be severely affected by the frequency of the location calibration of the AUV. Additionally, since it may take long service delays, AUV might have to stay submerged for extended periods of time adding to fuel and maintenance costs. Long service delays of AUV will also result in long localization delays for the nodes. Luo et al. [47] build on their work in [48] and mitigate the long AUV servicing and messaging delays by proposing a silent AUV-based scheme called LDB. AUVs are mounted with directional transceivers.

AUVs followed pre-planned trajectories while transmitting conical shaped directional acoustic beams at constant intervals. Nodes listen to these beacons silently and estimate their locations. Since nodes do not need to exchange messages with AUVs, AUVs can transmit long range directional beacons. This helps in greatly reducing the submerged stay of AUVs—resultantly reducing the fuel and maintenance costs. However, the scheme incorporates extra complexity in acoustic transceivers due to directional beacons. Furthermore, beacon transmission frequency also considerably affects the localization accuracy. Higher frequency achieves better estimation at the expense of beaconing overhead. A careful and practical tradeoff has to be achieved.

Advantages: High localization ratio and accuracy can be achieved. Trajectory planning could allow for complete localization coverage. Single anchor can transmit multiple unique beacons from different locations which eliminates the need for multiple anchors.

Disadvantages: Very high initial and maintenance cost. AUVs operation requires special training. Sophisticated trajectory planning, piloting and steering mechanisms need to be procured. AUVs cannot remain submerged for extended periods of time.

5.6 NDv (Fixed Anchors for AUV Localization)

AUVs act as beacon sources for many localization schemes. It is essential that AUVs maintain accurate estimate of their own position while submerged to avoid transmitting erroneous beacons. Most such schemes [26, 47] demand use of sophisticated trajectory tracking for AUVs while submerged. However, for improved accuracy, it is possible for AUVs to reconfirm their positions from fixed anchors while being submerged. Cheng et al. [16] and [17] propose a scheme called UPS which employs a hybrid of AUVs and fixed nodes for sensing purposes. By using four fixed anchors, AUVs and other sensing nodes can determine their position using trilateration-based ranging. The nodes detect differences in signal arrival times from four anchor nodes, and perform trilateration to determine their coordinates from range estimates. The scheme is silent and does not require synchronization. Authors take into account different error sources including receiver system delay, underwater multipath fading and variable acoustic speed underwater. UPS can have relevant applications for other schemes also where AUVs act as anchors, and can help increase accuracy of self-position estimate of AUVs. UPS however requires all the nodes to be within optimum range of four anchors which can be a design limitation in practical systems. The error incurred during ranging could be signified if the anchors are too close. Tan et al. [83] extends UPS in a scheme named WPS and proposes a variant which achieves localization with high probability regardless of positioning of reference nodes. WPS uses an extra anchor (UPS-5) to accomplish this higher localization ratio and accuracy. Not surprisingly, WPS has higher localization success at the expense of more messaging (beaconing) and higher anchor

density (extra anchor). Although not directly based on UPS and WPS, LSLS proposed by Cheng et al. [19] is considerably similar in implementation to these schemes and uses non-synchronization-based time difference (TDoA) measurements from anchors. The scheme computes time differences by employing local timers which eliminates the need for synchronization. The scheme is supplemented by iterative phases which select newly localized nodes as reference nodes. Each subsequent iterative phase is designed to increase accuracy and localization coverage. Although these schemes are not specifically designed for AUV localization, the similarities in their architecture and ranging methods demand them to be grouped together. Furthermore, UPS and WPS incorporate channel modeling for ocean current and error analysis due to acoustic signal fading. The model helps greatly increase the accuracy and reliability in location estimation for all sensing nodes including AUVs (if employed). Apart from providing localization to conventional stationary sensing nodes, these localization schemes can have applications in navigational assistance, collision avoidance, and steerage through obstacle prone submerged environments for AUVs.
Advantages: Assisted navigation support for AUVs. Schemes can be used for increased accuracy in self-position estimation for AUVs.
Disadvantages: Extra cost incurred. No dedicated system present at the moment.

5.7 Practical Systems

It is essential to survey practical 3D WSNs existing today so as to put these localization schemes into perspective. Presently, examples of 3D WSNs schemes (including localization) making beyond the realms of simulation environment into real world practical applications are rare. MASL is one of few 3D localization schemes tested on actual hardware [41, 91]. Usually inhibitive hardware costs and requirements for special test environments make physical testing difficult. Furthermore, most of the present proposed schemes are designed with little regard for physical world constraints and economic ramifications. In this section (Table 5) we present some practical 3D WSNs systems and their localization mechanisms. It is interesting to note the small scale of these networks. This highlights the requirement for academic researchers to design small, resilient and practical 3D localization schemes as opposed to most present large, complex schemes.

6 Major Work Summaries

This section lists the major works on Localization in 3D WSNs. Table 6 categorizes the work based on distributed/centralized, ranging mechanism, architecture and relevant applications.

Table 5 Practical 3D WSNs existing today and their localization procedures

System	Application	Localization procedure	Scale
Actuated acoustic sensor network [37]. Developer: University of Manchester	Monitoring nuclear waste storage pools	Small robot localized using acoustic beacons from anchors located at corners of pool	Small. Lab Environment. Manchester
Autonomous ocean sampling network [38]. Developer: MBARI	Deep ocean monitoring	Surface buoys, mobile anchors, deep sea nodes, AUVs act in coordination	Large. Since 2003. Monterey Bay. US
NASNet [39]. Developer: Nautronix (Commercial)	'Underwater GPS' system for positioning and navigational solutions	Assists in obtaining precise navigational data utilizing a subsea acoustic transmission grid	Large. Commercial solution
AMOUR (Autonomous Modular Optical Underwater Robot) [36]. Developer: CSAIL, MIT	Underwater monitoring. Research	Localization achieved using acoustic signals. High accuracy within 2.5 m	Small. Lab Environment
Seaweb [67] Developer: US Navy	Multiple. Anti submarine warfare, navigation, monitoring, intrusion detection, communication	Surface and submerged beacons. AUVs, Airborne, surface and submerged vehicles coordinate for acoustic ranging. Both long and short ranging	Large. Defense Applications. Multiple deployments since 1998. Large trials since 2003
Robotic Fish developer: Chonnam National University [77]. Michigan State University [75]	Academic Research. Environment monitoring	Submerged AUV not driven by propellers. GPS or RF TDoA-based systems employed. Localization occurs when AUV is at surface	Small. Lab Environment

7 Future Research Problems

This section discussed some of the future research problems for Localization in 3D Wireless Sensor Networks.

1. Most algorithms at the moment—especially for submerged 3D WSNs—employ anchors which have limited or full support for mobility. For most of these schemes

Table 6 Major works on localization in 3D WSNs

Scheme	Category	Type	Architecture	Anchor features	Ranging	Silence feature	Computation	Operational cost	Solution time	Relative practicality	Resilience or localization accuracy	Prospective applications	Major drawback
MASL [12, 54]	Submerged	Centralized	Mobile	No Anchors	ToA	No	Low	Medium	High	Medium	High	Monitoring, current studies	Synchronization requirements
3D-MALS [104]	Submerged	Centralized	Mobile	Mobile	ToA	No	Low	High	Medium	Medium	Medium	Monitoring	Redundant messaging, special hardware
CL [53]	Submerged	Centralized	Mobile	No anchors	ToA	No	Medium	Very high	High	Low	Medium	Deep sea monitoring	Synchronization, architecture dependent, node density requirements
AAL [26]	Submerged	Distributed	Hybrid	AUV	ToA	Yes	Low	Medium	Medium	Medium	High low	Coarse monitoring at low depths	Special hardware for pressure measuring, error increases at increased depths
LDB [47, 48]	Submerged	Distributed	Hybrid	AUV	Range-free	Yes	Low	Low	High	Low	Low low	Coarse monitoring at for high node density networks	AUV movement restricted to certain planes which might be impractical
MSL [25]	Submerged	Distributed	Mobile	Mobile and reference	ToA	No	High	High	Medium	High	High high	Defense, accurate measurements	Very high cost but practical solution
MSL [25]	Submerged	Distributed	Mobile	Mobile and reference	ToA	No	High	High	Medium	High	High high	Defense, accurate measurements	Very high cost but practical solution
LSHL [102]	Submerged	Distributed	Stationary	Underwater anchors and surface buoys collaborate	ToA	No	High	High	High (multi stage)	Medium	High	Large scale WSNs	Huge computational and messaging overhead

(continued)

Table 6 (continued)

Scheme	Category	Type	Architecture	Anchor features	Ranging	Silence feature	Computation	Operational cost	Solution time	Relative practicality	Resilience or localization accuracy	Prospective applications	Major drawback
DETL [15]	Submerged	Distributed	Mobile	Three anchors and reference nodes	ToA	No	High	High	High (multi stage)	Medium	High	Large scale WSNs	Huge computational and messaging overhead
3DUL [40]	Submerged	Distributed	Hybrid	No anchors, neighbors share estimation	ToA	No	High	Medium	High	Low	Medium low	Research, Low cost applications	Huge delays
AFL [57]	Submerged	Distributed	Stationary	No anchors	ToA likely	No	High	Low	High	High	Medium	Low cost delay tolerant monitoring	Huge delays, high node density requirements
UPS [16, 17]	Submerged	Distributed	Stationary	Four Stationary anchors	TDoA	Yes	Low	Low	Medium	High	Medium	Defense, Silent Applications	High node density requirements, localization might fail
WPS [57]	Submerged	Distributed	Stationary	1+ UPS anchors	TDoA	Yes	Low (UPS+)	Low (UPS+)	Medium	High	Medium	Improvement over UPS	High node density requirements, localization might fail
LSLS [83]	Submerged	Distributed	Stationary	Stationary Anchors	TDoA	No	Low (UPS+)	Low (UPS+)	Medium	High (UPS++)	High (UPS++)	Improvement over UPS	High node density requirements, localization might fail.
USP [18, 85, 86]	Submerged	Distributed	Stationary	Stationary anchors	Not specified	No	High	Medium	Medium	Low	Low	Research	Added Hardware, Synchronization
SLMP [22]	Submerged	Distributed	Mobile	Underwater anchors and surface buoys collaborate	ToA	No	High	Medium	Medium	High	High	Shallow water monitoring	Mobility support and added hardware
CL [14, 90]	General	Distributed	Stationary	Stationary Anchors	Range-free	No (potential yes)	High	Medium	Medium	Medium	Medium medium	Good for general applications (average accuracy)	Increased computation

(continued)

Table 6 (continued)

Scheme	Category	Type	Architecture	Anchor features	Ranging	Silence feature	Computation	Operational cost	Solution time	Relative practicality	Resilience or localization accuracy	Prospective applications	Major drawback
3D-CA [44]	General	Distributed	Stationary	Stationary Anchors	Range-free	No	High	Medium	High	Medium	Medium medium	Good for general applications (better accuracy than CL)	Increased computation
DV-HOP [46]	General	Distributed	Stationary	Stationary Anchors	Range-free	No	High	Medium (CL –)	High (CL ++)	High	Medium	General application	Increased delay and energy consumption
DHA [78]	General	Distributed	Stationary	Stationary Anchors	Range-free	No	Low	Low	Medium	High	Low	Low cost hardware specific applications	Prone to localization failure for irregular node placement
SSO [49]	General	Distributed	Stationary	Stationary Anchors	Range-free	No	Low	Low	Medium	High	Low	Low cost hardware specific applications	Prone to localization failure for irregular node placement
NSL [101]	General	Distributed	Stationary	Mobile nodes	Range-free	No	High	High	High	Low	High high	Interesting theoretical concept	Special hardware, very expensive
MSG [95]	General/ airborne	Distributed	Stationary	Mobile nodes	Range-free	No	Medium	High	Medium	Medium	High	Mobility Based	Expensive mobility support mechanism required.
FA [58]	General/ airborne	Distributed	Stationary	Mobile nodes	Range based	No	High	High	High	Medium	Medium	Airborne applications, defense	Expensive setup required
3D-AML [43]	General	Distributed	Stationary	Both applicable	Range based	No	High	High	High	Medium	Medium	general application	Increased computation
DODS [23]	General	Distributed	Stationary	Both applicable	Range based	No	Low	Low	Medium	High	Low low	Low cost low resilience applications	Prone to errors in some scenarios
3DT [50]	General	Distributed	Stationary	Stationary Anchors	Range based	No	High	Low	Medium	Medium	Medium	General application	Prone to errors in some scenarios

(continued)

Table 6 (continued)

Scheme	Category	Type	Architecture	Anchor features	Ranging	Silence feature	Computation	Operational cost	Solution time	Relative practicality	Resilience or localization accuracy	Prospective applications	Major drawback
MIC [52, 74]	General/ airborne	Distributed	Mobile	Mobile anchors	Range based	No	High	High	High	Low/ medium	High medium	Defense, expensive sensitive applications	High cost and advanced hardware
AMA [97]	General/ airborne	Distributed	Mobile	Mobile anchors	Range based	No	High	High	High	Low	High	Defense, expensive sensitive applications	High cost and advanced hardware
Landscape [99]	Landscape	Distributed	Stationary	Hybrid	Range based	No	Medium	High	Medium	Medium	Medium high	3D Terrain Networks	High cost and advanced hardware

mobile anchor plays a central role for submerged 3D localization. However there can be many practical scenarios when mobility might be hindered or not feasible. This might result in failure for node localization. Research needs to be focused on design of schemes which are adaptable and flexible and does not rely solely or heavily on submerged mobile anchors.

2. Most present schemes tend to achieve localization in an economically cost effective solution. However most schemes fail to incorporate the economic cost of mobile anchors both in terms of hardware and operational costs. Comparative analysis needs to be performed to weigh the potential gain/loss of such schemes. For example, deployment and operational cost analysis of fully mobile anchors versus increased density of static anchors or DETs can be done.
3. Few works have been done which incorporate speed of mobile anchors and path trajectories especially for 3D WSNs. There hardly exists any work which addresses the issue of path planning and trajectory of mobile anchors for 3D WSNs. Research work in this direction would be meaningful contribution in this field.
4. In submerged networks, correlated motion of the underwater nodes which due to ocean currents may be utilized favorably to provide assistance in localization as discussed in [28]. Research work in this area which utilizes natural energy of ocean currents for efficient localization would be an interesting contribution. However such work would need complex mathematical modeling and would need to incorporate seasonal variations etc.
5. Most localization algorithms assume that localization is being carried out independently of other network function. Cross layer optimization and integrating/merging of localization schemes in other network functions could result in serious performance gains.
6. It is explained in [61] that most of the work is directed towards optimization of localization algorithm and less work has been done in improvement of localization estimation. Accordingly, bias and variance performance are often secondary concerns. Reporting both bias and variance performance along with the Cramér-Rao lower bound (CRLB) will help provide a reference for comparison [11, 61]. Refer to [61] for more details on this problem.
7. Most present schemes are either range-based or range-free. Both schemes have their own pros and cons. However the distinction between the two schemes should not be as distinct as it is usually referred in literature. Instead research should be focused on proposing schemes which are hybrid of range-free and range-based and are adaptable and flexible according to application requirements. Such schemes would have the capacity to tune to desired accuracy mode.

8 Conclusion

In this survey we have tried to list all the major works carried out in the field of 3D localization in WSNs. The localization schemes presented are divided into two

sections: generic and submerged. The relative strengths and weaknesses of each scheme are also discussed. The survey can be categorized in three major sections. First section covers the classification and taxonomy. The second section can be considered the core of survey and lists most of 3D localization schemes and their prospective benefits. Third section outlays future and open problems in this field. The target of this survey is to provide the reader with basic understanding and quick reference of major efforts being carried out in recent past in the field of 3D localization in WSNs.

Acknowledgments The authors gratefully acknowledge the insightful comments of the anonymous reviewers which helped improve the quality and presentation of the paper significantly. This work is partially supported by the US National Science Foundation (NSF) grant 1054935 and 1224628.

References

1. I.F. Akyildiz, D. Pompili, T. Melodia, Underwater acoustic sensor networks: research challenges. Ad Hoc Netw. **3**(3), 257–279 (2005)
2. I. Amundson, X.D. Koutsoukos, A survey on localization for mobile wireless sensor networks, in *Proceedings of the 2nd International Conference on Mobile Entity Localization and Tracking in GPS-Less Environments (MELT'09)* (2009)
3. I. Amundson, X.D. Koutsoukos, A survey on localization for mobile wireless sensor networks, in *Proceedings of the 2nd International Conference on Mobile Entity Localization and Tracking in GPS-Less Environments, MELT'09*, ed. by R. Fuller, X.D. Koutsoukos (Springer, Berlin, 2009), pp. 235–254
4. P. Bahl, V.N. Padmanabhan, RADAR: an in-building RF-based user location and tracking system, in *Proceedings of the IEEE INFOCOM'00*, March 2000
5. J. Beutel, Geolocation in a picoradio environment, Master's thesis, UC Berkeley, Department of Electrical Engineering and Computer Science, 120p, 1999
6. A. Boukerche, H.A. B.F. Oliveira, E. Nakamura, A.A.F. Loureiro, Localization systems for wireless sensor networks. IEEE Wirel. Commun., December 2007
7. N. Bulusu, J. Heidemann, V. Bychkovskiy, D. Estrin, Density-adaptive beacon placement algorithms for localization in ad hoc wireless networks, in *IEEE Infocom 2002*, June 2002
8. N. Bulusu, J. Heidemann, D. Estrin, GPS-less lowcost outdoor localization for very small devices. IEEE Wirel. Commun. **7**(5), 28–34 (2000)
9. J. Caffery, G.L. Stuber, Subscriber location in CDMA cellular networks. IEEE Trans. Veh. Technol. **47**, 406–416 (1998)
10. M. Campbell, Intelligence in three dimensions: we live in a 3-d world, and so should computers. http://www.neptec.com/News2006/1Oct06-MilAero.html (2006)
11. Y.T. Chan, K.C. Ho, A simple and efficient estimator for hyperbolic location. IEEE Trans. Signal Processing **42**(8), 1905–1915 (1994)
12. T. Chance, A. Kleiner, J. Northcutt, *The Autonomous Underwater Vehicle (AUV): A Cost-Effective Alternative to Deep-Towed Technology*, 6th edn, Integrated Coastal Zone Management, 2000, pp. 65–69
13. Y.T. Chan, H.Y.C. Hang, P.C. Ching, Exact and approximate maximum likelihood localization algorithms. IEEE Trans. Veh. Technol. **55**(1), 10–16 (Jan. 2006)
14. H. Chen, P. Huang, M. Martins, H. Cheung So, K. Sezaki, Novel centroid localization algorithm for three-dimensional wireless sensor networks, in *4th International Conference on Wireless Communications, Networking and Mobile Computing, WiCOM'08*, vol., no., pp. 1–4, 12–14 Oct. 2008. doi:10.1109/WiCom.2008.841

15. K. Chen, Y. Zhou, J. He, A localization scheme for underwater wireless sensor networks. Int. J. Adv. Sci. Technol. **4** (2009)
16. X. Cheng, H. Shu, Q. Liang, A range-difference based self positioning scheme for underwater acoustic sensor networks, in *WASA '07 Proceedings of the International Conference on Wireless Algorithms, Systems and Applications* (2007)
17. X. Cheng, H. Shu, Q. Liang, D.H. Du, Silent positioning in underwater acoustic sensor networks, in *Proceedings of 29th IEEE International Conference on Distributed Computing Systems Workshops*, Montreal, Quebec, Canada (2009)
18. W. Cheng, A.Y. Teymorian, L. Ma, X. Cheng, X. Lu, Z. Lu, Underwater localization in sparse 3d acoustic sensor networks, in *Proceedings of IEEE INFOCOM*, Phoenix, AZ, USA (2008), pp. 236–240
19. W. Cheng, A. Thaeler, X. Cheng, F. Liu, X. Lu, Z. Lu, Time-synchronization free localization in large scale underwater acoustic sensor networks, in *29th IEEE International Conference on Distributed Computing Systems Workshops, 2009. ICDCS Workshops'09*, 22–26 June 2009, pp. 80–87
20. K.W. Cheung, H.C. So, W.K. Ma, Y.T. Chan, Least squares algorithms for time-of-arrival-based mobile location. IEEE Trans. Signal Processing **52**(4), 1121–1130 (2004)
21. H.S.M. Coxeter, *Introduction to Geometry* (Wiley, New York, 1989)
22. J. Cui, Z. Zhou, A. Bagtzoglou, Scalable localization with mobility prediction for underwater sensor networks, in *Proceedings of Second Workshop on Underwater networks (WuWNet)*, Montreal, Quebec, Canada (2007)
23. E. Doukhnitch, M. Salamah, E. Ozen, An efficient approach for trilateration in 3D positioning. Comput. Commun. **31**(17), 4124–4129 (2008)
24. E. Elnahrawy, X. Li, R.P. Martin, The limits of localization using signal strength: a comparative study, in *Proceeding of Sensor and Ad Hoc Communications and Networks, IEEE SECON*, October 2004, pp. 406–414
25. M. Erol, L. Vieira, A. Caruso, F. Paparella, M. Gerla, S. Oktug, Multi stage underwater sensor localization using mobile beacons, in *Proceedings of the Second International Workshop on Under Water Sensors and Systems*, Cap Esterel, France, August 2008, pp. 25–31
26. M. Erol, L. Vieira, M. Gerla, Auv-aided localization for underwater sensor networks, in *Proceedings f the International Conference on Wireless Algorithms, Systems and Applications (WASA)*, Chicago, IL, USA (2007)
27. M. Erol, L. Vieira, M. Gerla, Localization with dive'n'rise(dnr) beacons for underwater acoustic sensor networks, in *Proceedings of Second Workshop on Underwater Networks (WuWNet)*, Montreal, Quebec, Canada (2007), pp. 97–100
28. M. Erol-Kantarci, H.T. Mouftah, S. Oktug, A survey of architectures and localization techniques for underwater acoustic sensor networks. IEEE Commun. Surv. Tutor. **13**(3), 487–502, Third Quarter 2011
29. P. Gober, A. Ziviani, P. Todorova, M. Dias de Amorim, P. Hunerberg, S. Fdida, Topology control and localization in wireless ad hoc and sensor networks. Ad Hoc Sens. Wirel. Netw. **1**, 301–321 (2005)
30. I. Guvenc, C.-C. Chong, A survey on TOA based wireless localization and NLOS mitigation techniques, IEEE Commun. Surv. Tutor. **11**(3), 107–124, 3rd Quarter 2009
31. G. Han, H. Xu, T.Q. Duong, J. Jiang, T. Hara, Localization algorithms of wireless sensor networks: a survey. J. Telecommun. Syst. (2011). doi:10.1007/s11235-011-9564-7
32. A. Harter, A. Hopper, P. Steggles, A. Ward, P. Webster, The anatomy of a context-aware application, in *Proceedings of MOBICOM '99*, Seattle, Washington (1999)
33. T. He, C. Huang, B.M. Blum, J.A. Stankovic, T. Abdelzaher, Range-free localization schemes for large scale sensor networks, in *Proceedings of the 9th Annual International Conference on Mobile Computing and Networking (MobiCom '03)* (ACM, New York, 2003), pp. 81–95
34. T. He, C. Huang, B. Blum, J. Stankovic, T. Abdelzaher, Range-free localization schemes in large scale sensor networks, in *MobiCom* (2003)
35. J. Hightower, G. Boriello, R. Want, SpotON: an indoor 3D location sensing technology based on RF signal strength, University of Washington CSE, Report #2000-02-02, February 2000

36. http://groups.csail.mit.edu/drl/underwater_robotics/amour/amour.html
37. http://www.aasn4ip.org.uk/
38. http://www.mbari.org/aosn/
39. http://www.nautronix.com
40. M.T. Isik, O.B. Akan, A three dimensional localization algorithm for underwater acoustic sensor networks. IEEE Trans. Wirel. Commun. **8**(9), 4457–4463 (2009)
41. J. Jaffe, C. Schurgers, Sensor networks of freely drifting autonomous underwater explorers, in *Proceedings of the 1st ACM International Workshop on Underwater Networks (WUWNet '06)* (ACM, New York, 2006), pp. 93–96
42. J. Kong, J. Cui, D. Wu, M. Gerla, Building underwater adhoc networks and sensor networks for large scale real-time aquatic applications, in *Proceedings of IEEE MILCOM*, Atlantic City, NJ, USA (2005)
43. G.S. Kuruoglu, M. Erol, S. Oktug, Three dimensional localization in wireless sensor networks using the adapted multi-lateration technique considering range measurement errors, in *GLOBECOM Workshops, 2009*, IEEE, pp. 1–5, Nov. 30 2009–Dec. 4 2009
44. X. Lai, J. Wang, G. Zeng, Distributed positioning algorithm based on centroid of three-dimension graph for wireless sensor networks. Syst. Simul. Technol. **20**(15), 4104–4111 (2008)
45. K. Langendoen, N. Reijers, in *Distributed Localization in Wireless Sensor Networks: A Quantitative Comparison. Computer Networks (Elsevier)*, special issue on Wireless Sensor Networks, August 2003, pp. 374–387
46. W.A.N.G. Li, Z.H.A.N.G. Jing, C.A.O. Dun, A new 3-dimensional DV-hop localization algorithm. JCIS **8**(6), 2463–2475 (2012)
47. H. Luo, Z. Guo, W. Dong, Y.Z.F. Hong, Ldb: localization with directional beacons for sparse 3d underwater acoustic sensor networks. J. Netw. **5**(1) (2010)
48. H. Luo, Y. Zhao, Z. Guo, S. Liu, P. Chen, L.M. Ni, Udb: using directional beacons for localization in underwater sensor networks, in *Proceedings 14th IEEE International Conference on Parallel and Distributed Systems (ICPADS '08)*, Melbourne, Victoria, Australia (2008), pp. 551–558
49. L. Lv, Y. Cao, X. Gao, H. Luo, in *Three Dimensional Localization Schemes Based on Sphere Intersections in Wireless Sensor Network* (Beijing Posts and Telecommunications University, Beijing, 2006), pp. 48–51
50. D.E. Manolakis, Efficient solution and performance analysis of 3-D position estimation by trilateration. IEEE Trans. Aerosp. Electron. Syst. **32**(4), 1239–1248 (1996)
51. G. Mao, B. Fidan, B.D.O. Anderson, Wireless sensor network localization techniques. Comput. Netw. **51**, 2529–2553, 10 July 2007. doi:10.1016/j.comnet.2006.11.018
52. S.T. Matale, A.M. Kasabe, Review of advance localization algorithms, ijarece sep 2012
53. D. Mirza, C. Schurgers, Collaborative localization for fleets of underwater drifters, in *Proceedings of IEEE Oceans*, Vancouver, BC (2007)
54. D. Mirza, C. Schurgers, Motion-aware self-localization for underwater networks, in *Proceedings of Third ACM International Workshop on Underwater Networks*, San Francisco, CA, USA (2008), pp. 51–58
55. D. Niculescu, B. Nath, Ad-hoc positioning system, in *IEEE GlobeCom* (2001)
56. D. Niculescu, B. Nath, in *Ad Hoc Positioning System (APS) using AoA, INFOCOM' 03*, San Francisco, CA (2003)
57. A.K. Othman, A.E. Adams, C.C. Tsimenidis, Node discovery protocol and localization for distributed underwater acoustic networks, in *Proceedings of the Adv. Int. Conf. on Telecommunications, Internet, Web Applications and Services (AICT-ICIW)*, Washington, DC, USA (2006), p. 93
58. C.-H. Ou, K.-F. Ssu, Sensor position determination with flying anchor in three dimensional wireless sensor networks, in *IEEE Transactions on Mobile Computing*, September 2008, pp. 1084–1097
59. A. Panwar, S.A. Kumar, Localization schemes in wireless sensor networks, in *2012 Second International Conference on Advanced Computing & Communication Technologies (ACCT)*, pp. 443–449, 7–8 January 2012

60. J. Partan, J. Kurose, B.N. Levine, A survey of practical issues in underwater networks, in *WUWNet* (2006)
61. N. Patwari, J.N. Ash, S. Kyperountas, A.O. Hero III, R.L. Moses, N.S. Correal, Locating the nodes. IEEE Signal Processing Magazine, July 2005
62. N. Patwari, R. O'Dea, Y. Wang, Relative location in wireless networks, in *Proceedings of IEEE Veh. Tech. Conf. (VTC)* (2001)
63. D. Pompili, T. Melodia, I.F. Akyildiz, Deployment analysis in underwater acoustic wireless sensor networks, in *Proceedings of the 1st ACM International Workshop on Underwater Networks, WUWNet '06*, Los Angeles, CA, USA (ACM, New York, NY, 2006), pp. 48–55
64. N.B. Priyantha, A. Chakraborty, H. Balakrishnan, The cricket location-support system, in *Proceedings of MOBICOM '00*, New York, August 2000
65. J. Proakis, E. Sozer, J. Rice, M. Stojanovic, Shallow water acoustic networks. IEEE Commun. Mag. (2001), pp. 114–119
66. V. Ramadurai, M.L. Sichitiu, Localization in wireless sensor networks: a probabilistic approach, in *Proceedings of ICWN 2003*, Las Vegas, NV, June 2003, pp. 275–81
67. J. Rice, D. Green, Underwater acoustic communications and networks for US navy's seaweb program, in *Second International Conference on Sensor Technologies and Applications*. IEEE, 2008, pp. 715–722
68. M. Salamah, E. Doukhnitch, An efficient algorithm for mobile objects localization. Int. J. Commun. Syst. 21 (3), 301–310 (2007)
69. C. Savarese, J.M. Rabaey, J.Beutel, Locationing in distributed ad-hoc wireless sensor network, in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, Salt Lake City, UT, May 2001, pp. 2037–2040
70. A. Savvides, C.C. Han, M.B. Srivastava, Dynamic fine grained localization in ad-hoc networks of sensors, in *Proceedings of MOBICOM '01, 2001*, Rome, Italy, July 2001
71. A. Savvides, C. Han, M. Strivastava, Dynamic fine-grained localization in ad-hoc networks of sensors, in *Proceedings of ACM SIGMOBILE 2001*, Rome, Italy, July 2001
72. A. Savvides, H. Park, M. Srivastava, The bits and flops of the n-hop multilateration primitive for node localization problems, in *First ACM International Workshop on Wireless Sensor Networks and Application*, September 2002
73. A. Savvides, H. Park, M.B. Srivastava, The n-hop multilateration primitive for node localization problems. Mob. Netw. Appl. **8**(4), 443–451 (2003)
74. Y. Shang, W. Ruml, Y. Zhang, M. Fromherz, Localization from mere connectivity, in *ACM MobiHoc* Annapolis, MD, June 2003, pp. 201–212
75. S. Shatara, X. Tan, E. Mbemmo, N. Gingery, S. Henneberger, Experimental investigation on underwater acoustic ranging for small robotic sh, in *2008 IEEE International Conference on Robotics and Automation*. IEEE (2008)
76. G. Shen; R. Zetik, R.S. Thoma, Performance comparison of TOA and TDOA based location estimation algorithms in LOS environment, in *5th Workshop on Positioning, Navigation and Communication, WPNC 2008*, pp. 71–78, 27–27 March 2008. doi:10.1109/WPNC.2008.4510359
77. D. Shin, S. Na, J. Kim, S. Baek, M. Song, A. Park, S. Korea, Development of a gps-based autonomous water pollution monitoring system using sh robots, in *Proceedings of the 6th WSEAS International Conference on Computational intelligence, Man-Machine Systems and Cybernetics* (World Scientific and Engineering Academy and Society (WSEAS), 2007)
78. J. Shu, L. Liu, Y. Chen, A novel three-dimensional localization algorithm in wireless sensor networks, wireless communications, networking and mobile computing, in *Proceedings of 5th International Conference on Wireless Communications*, (2009), pp. 24–29
79. S. Simic, S. Sastry, Distributed localization in wireless ad hoc networks, UC Berkeley, Tech. rep. UCB/ERL M02/26, 2002
80. E. Sozer, M. Stojanovic, J. Proakis, Underwater acoustic networks. IEEE J. Ocean. Eng. **25**(1), 72–83 (2000)
81. M. Stojanovic, in *Acoustic (Underwater) Communications*, ed by J.G. Proakis. Encyclopedia of Telecommunications (John Wiley and Sons, New York, 2003)

82. H.-P. Tan et al., A survey of techniques and challenges in underwater localization. Ocean Eng. **38**(14), 1663–1676 (2011)
83. H.-P. Tan, A.F. Gabor, Z.A. Eu, W.K. G. Seah, A wide coverage positioning system (wps) for underwater localization, in *Proceedings of IEEE International Conference on Communications (ICC)*, Cape Town, South, Africa, May 2010, pp. 1–5
84. X. Tan, J. Li, Cooperative positioning in underwater sensor networks. IEEE Trans. Signal Processing **58**(11), 5860–5871 (2010)
85. A.Y. Teymorian, W. Cheng, L. Ma, X. Cheng, An underwater positioning scheme for 3d acoustic sensor networks, in *Proceedings of the Second workshop on Underwater Networks (WuWNet)*, Montreal, Quebec, Canada (2007)
86. A.Y. Teymorian, W. Cheng, L. Ma, X. Cheng, X. Lu, Z. Lu, 3d underwater sensor network localization. IEEE Trans. Mob. Comput. **8**(12), 1610–1621 (2009)
87. F. Thomas, L. Ros, Revisiting trilateration for robot localization. IEEE Trans. Robot. **21**(1), 93–101 (2005)
88. H.L.V. Trees, *Detection, Estimation, and Modulation Theory—Part 1* (Wiley, New York, 1968)
89. UnderWater Sensor Networks at BWN Laboratory, Georgia Institute of Technology, Available from http://www.ece.gatech.edu/research/labs/bwn/UWASN/
90. Q. Wan, Y.N. Peng, An improved 3-dimensional mobile location method using volume measurements of tetrahedron. IEICE Trans. Commun. *E85-B*, 1817–1823 (2002)
91. S. Wang, H. Hu, Wireless Sensor Networks for Underwater Localization: A Survey, Technical Report: CES-521. University of Essex, Colchester CO4 3SQ, UK
92. M.K. Watfa, Practical applications and connectivity: algorithms in future wireless sensor networks. Int. J. Inf. Technol. **4**, 18–28 (2007)
93. B.H. Wellenhoff, H. Lichtenegger, J. Collins, *Global Positions System: Theory and Practice*, 4th edn (Springer, Berlin, 1997)
94. P. Xie, J. Cui, L. Lao, Vbf: vector-based forwarding protocol for underwater sensor networks, in *Proceedings of IFIP Networking*, Portugal (2006)
95. V. Yadav, et al., Localization scheme for three dimensional wireless sensor networks using GS enabled mobile sensor nodes. Int. J. Next Gen. Netw. **1**(1) (2009)
96. A. Youssef, M. Youssef, A taxonomy of localization schemes for wireless sensor networks, in *The 2007 International Conference on Wireless Networks, ICWN*, Nevada, US (2007), pp. 444–450
97. G. Yu, F. Yu, L. Feng, A three dimensional localization algorithm using a mobile anchor node under wireless channel, in *International Joint Conference on Neural Networks* (2008), pp. 477–483
98. G. Zanca, F. Zorzi, A. Zanella, M. Zorzi, Experimental comparison of RSSI-based localization algorithms for indoor wireless sensor networks, in *Proceedings of the Workshop on Real-World Wireless Sensor Networks (REALWSN '08)* (ACM, New York, 2008), pp. 1–5. doi:10.1145/1435473.1435475
99. L. Zhang, X. Zhou, Q. Cheng, Landscape-3D: a robust localization scheme for sensor networks over complex 3D terrains, in *Proceedings of 31st Annual IEEE Conference on Local Computer Networks (LCN)* (2006), pp. 239–246
100. Y. Zhao, H. Wu, M. Jin, S. Xia, Localization in 3D surface sensor networks: challenges and solutions, in *2012 Proceedings IEEE INFOCOM*, pp. 55–63, 25–30 March 2012 doi:0.1109/INFCOM.2012.6195798
101. S. Zheng, L. Kai, Z.H. Zheng, Three dimensional localization algorithm based on nectar source localization model in wireless sensor network. Appl. Res. Comput. **25**(8), 2512–2513 (2008)
102. Z. Zhou, J. Cui, S. Zhou, Localization for large-scale underwater sensor networks, in *Proceedings of IFIP Networking*, Atlanta, Georgia, USA, May, 2007, pp. 108–119
103. H. Zhou, H. Wu, S. Xia, M. Jin, N. Ding, A distributed triangulation algorithm for wireless sensor networks on 2D and 3D surface, in *Proceedings of INFOCOM* (2011)
104. Y. Zhou, B. Gu, J.C.K. Chen, H. Guan, An range-free localization scheme for large scale underwater wireless sensor networks. J. Shangai Jiaotong Univ. **14**(5), 562–568 (2009)

Part V
Topology Control and Routing in Three-Dimensional Wireless Sensor Networks

Chapter 10
Three-Dimensional Wireless Sensor Networks: Geometric Approaches for Topology and Routing Design

Yu Wang

Abstract Three-dimensional (3D) wireless sensor networks have attracted a lot of attention due to their great potential usages in both commercial and civilian applications, such as environmental data collection, pollution monitoring, space exploration, disaster prevention, and tactical surveillance. Unfortunately, the design of 3D networks is surprisingly more difficult than the design in two-dimensional (2D) networks. Many properties of the network require additional computational complexity, and many problems cannot be solved by extensions or generalizations of 2D methods. In addressing these challenges, there have been new network protocols and algorithms designed for 3D wireless sensor networks using geometric approaches. In this chapter, we review the most recent advances in 3D topology control and 3D geographic routing for 3D wireless sensor networks.

1 Introduction

Most existing wireless sensor network systems and protocols are based on two-dimensional design, where all wireless sensor nodes are distributed in a two-dimensional (2D) plane. This assumption is somewhat justified for applications where sensor nodes are deployed on earth surface and where the height of the network is smaller than transmission radius of a node. However, this 2D assumption may no longer be valid if a wireless sensor network is deployed in space, atmosphere, or ocean, where nodes of a network are distributed over a 3D space and the difference in the third dimension is too large to be ignored. In fact, recent interest in underwater sensor networks [3], underground sensor networks [4], airborne sensor networks [6], ocean sensor networks [98], or space sensor networks [33] hints at the strong need to design 3D wireless networks. Nevertheless, sensor network problems in 3D have

Y. Wang (✉)
University of North Carolina at Charlotte,Charlotte, NC 28223, USA
e-mail: yu.wang@uncc.edu

H. M. Ammari (ed.), *The Art of Wireless Sensor Networks*,
Signals and Communication Technology, DOI: 10.1007/978-3-642-40066-7_10,

not been adequately analyzed until recently. There is a tendency to either ignore the extension of algorithms from 2D to 3D or believe that it is straightforward. Unfortunately, the design of 3D networks is surprisingly more difficult than the design in 2D. Many properties of the network require additional computational complexity, and many problems cannot be solved by extensions or generalizations of 2D methods. In facing up to these challenges, there have been new network protocols and algorithms specifically designed for 3D wireless sensor networks using geometric approaches by exploring rich geometric properties of sensor networks. In this chapter, we will review the most recent advances in 3D wireless sensor networks with a focus on geometric approaches for two particular sensor network problems: *3D topology control* and *3D geographic routing*.

1.1 Applications of 3D Wireless Sensor Networks

Three-dimensional wireless sensor networks have a variety of applications. Figure 1 shows three possible applications of 3D wireless sensor networks.

Underwater sensor network: Underwater acoustic sensor network (UWSN) [3] or ocean sensor network [98], shown in Fig. 1a, can find applications in oceanographic data collection, pollution monitoring, offshore exploration, disaster prevention,

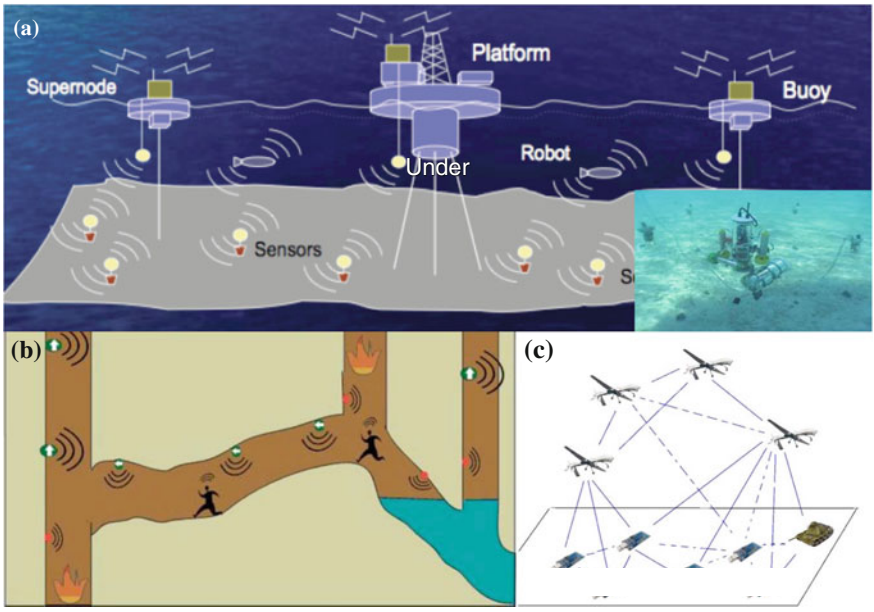


Fig. 1 Examples of three-dimensional wireless sensor networks: **a** underwater sensor network, **b** underground sensor network, and **c** airborne sensor network

assisted navigation, and tactical surveillance applications. Three-dimensional UWSN is used to detect and observe phenomena that cannot be adequately observed by means of ocean bottom sensor nodes (in a 2D sensor network), i.e., to perform cooperative sampling of the 3D ocean environment. In 3D UWSN, sensor nodes float at different depths to enable the exploration of natural undersea resources and gathering of scientific data in collaborative monitoring missions. Besides the 3D deployment, UWSN is also significantly different from terrestrial sensor networks: (1) acoustic channels used under water feature long propagation delays and low bandwidth; (2) underwater sensor nodes may move with water, thus introducing passive mobility.

Underground sensor network: Underground sensor network (UGSN) [4] can be used to monitor a variety of conditions, such as soil properties for agricultural applications, integrity of belowground infrastructures for plumbing, or toxic substances for environmental monitoring. For example, agriculture can use underground sensors to monitor soil conditions such as water and mineral content [21]. Wireless sensors can also be used to monitor the underground tunnels in coal mines [52] as shown in Fig. 1b. These tunnels are usually long and narrow and distributed in 3D, with lengths of tens of kilometers and widths of several meters. A full-scale monitoring of the tunnel environment (including the amount of gas, water, and dust) has been a crucial task to ensure safe working conditions in coal mines.

Airborne sensor network: Unmanned air vehicles (UAVs) has been proposed to be used as mobile, adaptive communication backbones for ground-based sensor networks [75, 86]. Figure 1c illustrates that multiple UAVs can serve as an airborne relay for a ground-based sensor networks. The UAVs and the sensor nodes together form a 3D airborne sensor network to support the military applications. The UAVs can also provide communication connectivity to sensors that cannot communicate with each other because of terrain, distance, or other geographical constraints. Moreover, the UAVs themselves can have sensing capacity and form a pure airborne sensor network [6].

Besides examples above, 3D wireless sensor networks can also be found useful in many other applications, such as a large 3D space network for space explorations [33] or a small 3D sensor network in a multi-floor building for structure monitoring [106]. Due to its wide-range potential applications, 3D wireless sensor network has recently emerged as a premier research topic in wireless sensor network community.

1.2 Model of 3D Wireless Sensor Networks: Unit Ball Graph

A 3D wireless sensor network consists of a set V of n wireless sensor nodes distributed in a 3D plane $\mathbb{R}^3$. Each sensor node has the same *maximum transmission range* R, thus its transmission region can be simply modeled as a 3D sphere with radius R (as shown in Fig. 2a). All wireless sensor nodes define a *unit ball graph* (UBG), or called *unit sphere graph*, as shown in Fig. 2b, in which there is an edge uv between two nodes u and v if and only if the Euclidean distance $\|uv\|$ between u and v in $\mathbb{R}^3$ is at most R. In other words, two nodes can always receive the sig-

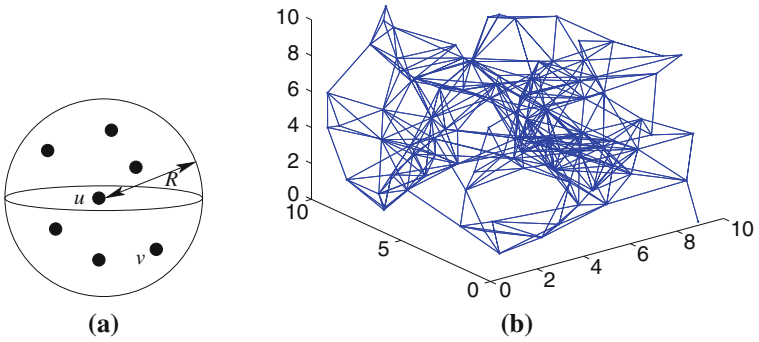


Fig. 2 Illustrations of unit ball graphs: **a** the transmission region around sensor u is modeled as a unit ball with radius R and **b** a unit ball graph formed by 100 sensors

nal from each other directly if the distance between them is not more than R. For simplification, we usually assume $R = 1$ in our analysis. If there exists a link uv in UBG, v is a neighbor of u. All neighbors of u form its one-hop neighborhood, denoted as $N_{UBG}(u)$ or $N(u)$. The size of $N_{UBG}(u)$ is the node degree of u in UBG. Clearly, UBG includes all possible communication links in a 3D wireless sensor network.

We assume that all wireless nodes have distinctive identities and each node knows its position information either through a low-power GPS receiver or some other ways (such as 3D localization methods in [19, 111, 114]). By one-hop broadcasting, each node u can gather the location information of all nodes within its transmission range. As in the most common power-attenuation model, the power to support a link uv is assumed to be $\|uv\|^{\beta}$, where β is the path loss exponent which is a real constant between 2 and 5 depending on the wireless transmission environment.

1.3 Topology Control and Routing in Wireless Sensor Networks

Topology control technique is to let each sensor node *locally* adjust its transmission range and select certain neighbors for communication, while maintaining a structure that can support energy efficient routing and improve the overall network performance. For a 3D wireless sensor network, given the UBG defined by all sensors, topology control aims to build and maintain a sparse 3D subgraph of the UBG as the underlying topology for the network. Unlike traditional wired networks, sensor nodes are often moving or changing status during the communication, which could change the network topology. Hence it is more challenging to design a topology control algorithm for sensor networks: the topology should be locally and self-adaptively maintained without affecting the whole network, and the communication cost during maintaining should not be too high. *Routing* is also a challenging task in wireless sensor networks, which aims to delivery packets from a source

node to a destination node via multihop relays. Route discovery in most existing routing protocols (for wired networks or even wireless ad hoc networks) can be very expensive in communication costs, thus reducing the response time of the network. On the other hand, explicit route maintenance can be even more costly in the explicit communication of substantial routing information and the usage of scare memory of wireless sensor nodes. Fortunately, the geometric nature of multihop wireless sensor networks allows a set of promising approaches to confront these challenges.

1.4 Geometric Approaches for Wireless Sensor Networks

In wireless sensor networks (especially 3D wireless sensor networks), sensors are distributed in a certain geographical region. The physical locations of sensor nodes greatly impact the system design. With rich geometric properties of the wireless sensor network, many geometric algorithms can be used to solve hard problems (such as localization, topology control, naming, and routing) and provide provable performance guarantee even in a probabilistic and dynamics world. In the past several years, geometric-structure-based topology control algorithms [32, 51, 58, 59, 72, 73, 77, 87, 103] have been proposed and widely used in sensor networks. They are primarily targeted to maintain network connectivity, optimize network throughput with power-efficient routing, conserve energy, and increase fault tolerance. At the same time, quite a few geographic routing protocols [17, 37, 41, 43, 45, 84, 94] have been proposed and adopted for ad hoc and sensor networks to improve scalability. Localized geographic routing protocols do not need the dissemination of route discovery information, and no routing tables are maintained at each node. They only use the local position information at each node and geometric properties of surrounding neighbors to determine how to route the packet. This leads to lower overhead and higher scalability, and makes such routing protocols suitable for large-scale and complex wireless sensor networks. In this chapter, we will see different new geometric approaches which are designed for topology control or geographic routing in 3D wireless sensor networks.

Chapter Organization: The rest of the chapter is organized as follows. We review the current 3D geometric approaches for energy efficient topology control and localized geographic routing in Sects. 2 and 3, respectively. In each section, we first briefly discuss the prior art for 2D networks and new challenges in 3D networks, and then present current solutions in details. Finally, we conclude this chapter with a short summary in Sect. 4.

2 Topology Control for 3D Wireless Sensor Networks

In this section we focus on the design of 3D geometric topologies which support energy-efficient paths and/or fault tolerance for communications while can be easily constructed with purely local information.

2.1 Topology Control Problem for Energy Efficiency

Topology control protocols aim to maintain a 3D structure H from the original communication graph UBG of a 3D sensor network that can preserve connectivity, optimize network throughput with power-efficient routing and conserve energy. The constructed 3D topology H could be a directed or undirected graph. In the literature, the following desirable features of the structure are well-regarded and preferred in wireless sensor networks:

(1) **Connectivity:** To guarantee communications among all sensor nodes, the constructed topology H needs to be *connected*, i.e., there exists a path between any pair of nodes in the topology. This is the most fundamental requirement of topology control. Here, we always assume that the original communication graph UBG is a connected graph.
(2) **Bounded Node Degree:** It is also desirable that node degree in the constructed topology H is small and upper-bounded by a constant. If H is a directed graph, both in-degree and out-degree should be bounded. A small node degree reduces the MAC-level contention and interference, and may help mitigate the well-known hidden and exposed terminal problems. In addition, if a graph has a bounded node degree, it is also a sparse graph, i.e., the total number of links is linear with the total number of nodes in the graph. A sparse graph conserves more energy in term of maintaining the constructed network topology.
(3) **Energy Spanner:** A good network topology should be *energy efficient*, i.e., the total energy consumption of the least energy cost path between any two nodes in final topology should not exceed a constant factor of the energy consumption of the least energy cost path in the original network [58]. Given a path $\nu_1\nu_2\cdots\nu_h$ connecting two nodes ν_1 and ν_h, the energy cost of this path is $\sum_{j=1}^{h-1}\|\nu_j\nu_{j+1}\|^{\beta}$. The path with the least energy cost is called the shortest path in a graph. A subgraph H is called a *energy spanner* of a graph G if there is a positive real constant ρ such that for any two nodes, the energy consumption of the shortest path in H is at most ρ times of the energy consumption of the shortest path in G. The constant ρ is called the *energy stretch factor*. An energy spanner of the communication graph (e.g., UBG) keeps the possibilities of energy-efficient routing.
(4) **Fault Tolerance:** Due to constrained power capacity, hostile deployment environment, and other factors, events like individual node failures are more likely to happen in real life wireless sensor networks. To achieve fault tolerance

(such as surviving $k-1$ node failures), the constructed topology needs to be k-connected, given the communication graph (e.g., UBG) is k-connected.

(5) **Localized Construction:** Due to limited resources and high mobility of wireless nodes, it is preferred that the topology can be constructed locally and in a self-organizing fashion. Here, a topology is *localized*, i.e., can be constructed locally, if every node u can decide all edges incident on itself in the topology by only using the information of nodes within a constant hops of u. Actually, all construction algorithms of our topologies presented here only use 1-hop neighbor information. Notice that if multi-hop neighboring information or global information is available, some distributed or centralized algorithms can be applied.

2.2 Previous Solutions for 2D Sensor Networks

With the objective of achieving energy efficiency and maintaining network connectivity, several localized geometrical structures have been proposed for topology control in 2D wireless networks, such as *local minimum spanning tree* (LMST) [54, 62], *relative neighborhood graph* (RNG) [16, 81], *Gabriel graph* (GG) [16, 37], *Yao graph* (YG) [58, 59], *cone-based topology control* (CBTC) [51, 103], *Delaunay-based graph* (Del) [29, 55, 57], and different combinations of these graphs [56, 82, 92]. By constructing such sparse topology structures, transmission power of nodes can be minimized. As a result, the number of links in the constructed topology is significantly reduced compared with that of the original communication graph which contains all links supported by the maximum transmission power (modeled by the *unit disk graph* (UDG) in 2D, which contains an edge between two nodes if and only if their distance is at most one). Among these 2D structures, some are planar structures (such as LMST, RNG, GG and Delaunay-based graphs), some are energy spanners of UDG (such as GG, Yao graph, CBTC, and Delaunay-based graphs), and some are with bounded node degree (such as Yao graph). On the other hand, lacking of redundancy makes the topology more susceptible to node failures or link breakages. In order to achieve routing redundancy and construct k-connected topologies, existing topology control algorithms have been extended to incorporate fault tolerance (such as FLSS_k[53], CBTC_k[11], RNG_k/GG_k[115], $\text{YG}_{p,k}$[60]).

Besides these localized geometrical structures, there are also other various techniques proposed by researchers for topology control in 2D sensor networks, such as how to construct a virtual backbone for routing [7, 14, 63, 69] and how to minimize the total transmission power while maintaining connectivity or other properties [20, 65, 93, 116, 117]. We refer readers to some nice surveys [72, 77, 87] on 2D topology control for more details.

2.3 Localized 3D Topology Control for Energy Efficiency

Although geometric topology control protocols have been well studied in 2D networks, the design of 3D topology control is surprisingly more difficult than the design in 2D. Current 2D methods cannot be directly applied in 3D networks. There is no embedding method mapping a 3D network into a 2D plane so that the relative scale of all edge length is preserved and all 2D geometric topology control protocols can still be applied for energy efficiency. In other words, any simple mapping method from 3D to 2D does not work for energy efficient topology control.

Simply mapping from 3D to 2D does not work for topology control: For 3D topology control, the first thought is whether there exists an embedding method mapping the 3D networks onto a 2D plane so that all 2D geometric topology control protocols can still be applied. Here, an embedding of a 3D graph $G = (V, E)$ into a 2D plane is a mapping $f : V \to \mathbb{R}^2$, i.e., each vertex $v_i \in V$ with 3D position is identified with a point v_i' in the 2D plane. To keep the spanner property of 2D geometric topologies in 3D, we need to have a mapping method f such that for any two vertex v_i and v_j the distances between them in 2D and 3D plane should satisfy $\|v_i'v_j'\|_{2D} = a\|v_iv_j\|_{3D}$ where a is a constant. By applying such a mapping method, for any UBG G in 3D with a transmission range R, we can define a G' in 2D by scaling the transmission range to aR. The G and G' share the same connectivity, node degree, and stretch factors. Unfortunately, there is no such mapping method. Consider a regular tetrahedron (see Fig. 3) where the distances between any two nodes from the four endpoints are the same. In a 2D plane, there is no graph formed by four nodes where the distances between any two nodes are the same. Therefore, mapping 3D networks to 2D ones does not work when we want to achieve energy efficiency.

Therefore, there have been many new 3D geometric structures [11, 31, 38, 50, 90, 91] proposed for 3D wireless sensor networks. Next, we review them by groups. We start with the group of sparsest 3D structures (LMST, 3D GG and 3D RNG), then discuss a group of 3D structures with bounded degree (3D Yao graph and its relatives), and finally introduce 3D Delaunay, which is the most complicated structure and hard to built with only local information.

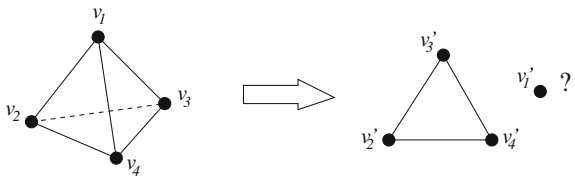


Fig. 3 Mapping 3D into 2D does not work for topology control: no such mapping maintaining the relationship of lengths among links exists

2.3.1 LMST, 3D GG, and 3D RNG

The *minimum spanning tree*, denoted by MST, is the tree which connects all nodes with minimized total edge length. MST is obviously one of the sparsest possible connected subgraphs, but its energy stretch factor can be as large as $n - 1$ and it cannot be built locally. Therefore, Li et al. [54] proposed a localized topology called *local minimum spanning tree* (LMST) to estimate MST and keep the network connected. In LMST, each node builds its local minimum spanning tree in one-hop neighborhood independently and only keeps one-hop on-tree nodes as neighbors. They proved that LMST is connected, and has a bounded degree of 6. Then, Li et al. [62] further proposed *k-localized minimum spanning tree* (LMST_k) which can be locally constructed using k-hop information. LMST_k is connected and planar, the total edge length of the LMST_k is within a constant factor of that of the MST when $k \geq 2$. Notice that LMST (or LMST_k) still works for 3D sensor networks. It can guarantee the connectivity of the network and is easy to be constructed by using local information. However, as the same problem as in 2D, the energy stretch factor of LMST can be arbitrarily large, i.e., unicast routing between a pair of nodes may need to travel a much longer distance than the shortest path in UBG.

It is very natural to extend the *related neighborhood graph* (RNG) [85] and *Gabriel graph* (GG) [26] to 3D. The definitions of 3D RNG and 3D GG are as follows: an edge $uv \in$ RNG if and only if the intersection of two balls centered at u and v with radius $\|uv\|$ does not contain any node from the set v; an edge $uv \in$ GG if and only if the ball with edge uv as a diameter contains no other node of v. Figure 4 illustrates the new 3D definitions. RNG and GG contains MST, which indicates that they are connected if the UBG is connected. However, both of them do not have bounded node degree. Assume that a node u has large number of neighboring nodes on the surface of its transmission sphere. In both RNG and GG, these neighboring nodes are all kept by u, thus u will have a large node degree. From the definitions, RNG $\subseteq$ GG since the shaded area in GG is included in the one in RNG. By using [58]'s example and proof in 2D case, we can prove that RNG is not an energy spanner, while GG is an energy spanner with the energy stretch factor of one. In other words, all edges in the least energy path in UBG are kept in GG.

Based on their definitions, 3D RNG and 3D GG can be easily constructed using 1-hop neighbors' position information only. Thus, the message complexity of the topology control protocol is $O(n)$ where n is the total number of nodes. Notice that all of these message exchanges are for learning the positions of 1-hop neighbors. There is no additional message exchange needed after each node learns its 1-hop

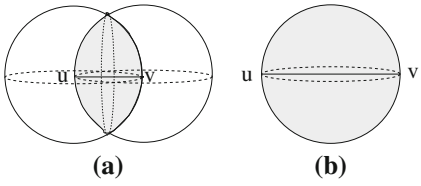


Fig. 4 Definitions of **a** 3D RNG and **b** 3D GG. *Shaded areas* contain no nodes

neighbors via periodic beacon messages. To check whether a neighbor w is inside the shaded areas, node u can simply check whether $\|uw\| < \|u\nu\|$ and $\|w\nu\| < \|u\nu\|$ for RNG or $\|uw\|^2 + \|w\nu\|^2 < \|u\nu\|^2$ for GG. Therefore, the time complexity of the topology control protocol at each node is $O(d)$ where d is the number of neighbors at that node.

2.3.2 3D Yao Structures and CBTC-Based Methods

Since both 3D RNG and 3D GG cannot bound node degree, it is also of interest to extend Yao graph to 3D. *Yao graph* [107] is originally proposed for construction of high-dimensional MST, but has been recently used for topology control in 2D ad hoc and sensor networks [58, 59]. In 2D, Yao graph YG_k is defined as follows. At each node u, any k equally-separated rays originating at u define k cones. In each cone, choose the shortest edge uv among all edges emanated from u, if there is any, and add a directed link $\overrightarrow{uv}$. In [58, 59], Li et al. proved that 2D Yao graph is an energy spanner of UDG. Their proof of the spanner property is based on the induction of link length. As shown in Fig. 5, for each removed link $uv \notin YG_k$, they proved the energy consumed by a shorter link uw and a path from w to v is within constant times of the energy consumed by link uv. The key result of their proof can be summarized by the following lemma:

Lemma 1. *[58, 59] The energy stretch factor of the Yao-based graph is at most* $\frac{1}{1-(2\sin\frac{\delta}{2})^\beta}$*, if for every link uv that is not in the final graph, there exists a shorter link uw in the graph and* $\angle vuw < \delta$*, where* δ *is a constant smaller than* $\pi/3$*, as shown in Fig. 5.*

Clearly, if $k > 6$, 2D Yao graph has its energy stretch factor bounded by $\frac{1}{1-(2\sin\frac{\pi}{k})^\beta}$.

3D Yao structures can use certain types of 3D cones to partition the transmission region of a node (which is a sphere), and inside each 3D cone the node only keeps a link to the nearest neighbor. If the number of such 3D cones is bounded by a constant k, 3D Yao structures can bound the node out-degree by k. However, it is hard to define the partition boundary of Yao structure of a node in 3D. Notice that a disk in 2D can be easily divided into k equal 2D cones which do not intersect with each other, but in 3D case, it is hard to divide a sphere into k equal 3D cones without intersections among each other. Basically, 3D Yao structures can be categorized into two sets based on their partition methods: fixed partition and flexible partition.

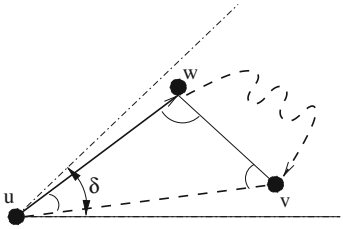


Fig. 5 Illustration of Lemma 1 for spanner property of Yao structure

3D Yao Structures based on Fixed Partition:

In fixed partition, 3D cones from one node do not intersect with each other and the partition method is the same for all nodes. In [90, 91], Wang et al. first proposed two methods to divides the transmission range of a node into certain number of 3D cones. Figure 6a and b illustrate these two methods, which divide the transmission ball into 32 and 56 cones respectively. For each cone, node u will choose the shortest edge $uv \in UBG$ among all edges emanated from u, if there is any, and add a directed link $\overrightarrow{uv}$. Ties are broken arbitrarily or by ID. The resulting directed graphs are denoted by $FiYG_{32}$ and $FiYG_{56}$ respectively. Notice that these cones in $FiYG_{32}$ and $FiYG_{56}$ are different and do not intersect with each other.

In $FiYG_{32}$, for each node u, it first divides its transmission region into 8 pieces by three orthogonal planes (i.e., xy-plane, yz-plane and zx-plane), where each piece is a $1/8$ sphere. Then it uses three more planes as shown in Fig. 6a to cut each piece into four cones (nodes c_1, c_2 and c_3 are the middle points of arcs in the $1/8$ sphere). In $FiYG_{56}$, smaller cones are used for partition, as shown in Fig. 6b. Each piece of the $1/8$ sphere is cut into 7 cones, where c_1 and c_1', c_2 and c_2', c_3 and c_3' trisect the arcs of $1/8$ sphere respectively.

Recently, Kim et al. [38] proposed another localized Yao-based structure with Platonic solid (PYG) which also uses a fixed partition method. To construct PYG, each node divides the 3D sphere neighborhood into k equal cones by using a regular k-polyhedron and selects the nearest neighbor in each cone. The resulting directed graph is denoted by PYG_k. Possible polyhedrons include tetrahedron, cube, octahedron, dodecahedron, and icosahedron for $k = 4, 6, 8, 12, 20$ respectively. Figure 6c and d illustrates partition examples with an octahedron $k = 8$ and a dodecahedron $k = 12$. Notice that the cones in this method are with same shape/size and do not intersect with each other.

For all these 3D Yao structures based on fixed partition ($FiYG_k$ or PYG_k), it is obvious that they have a bounded out-degree of k. In terms of energy spanner, it

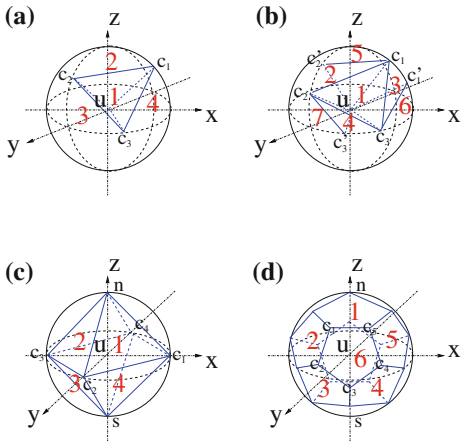


Fig. 6 Definitions of 3D Yao Structures with fixed partitions: **a** and **b** partitions of the $1/8$ sphere in FiYG: $FiYG_{32}$ and $FiYG_{56}$; **c** and **d** partitions using an octahedron or a dodecahedron for PYG: PYG_8 and PYG_{12}

depends how large k is used during the construction. For example, it is easy to show that the largest angle inside a cone in $FiYG_{32}$ is the angle $\angle c_1uc_2 = \pi/3$. Thus, the energy stretch factor $\frac{1}{1-(2sin(\frac{\delta}{2}))^{\beta}}$ in Lemma 1 could be infinite when $\delta = \pi/3$. Similar situation happens for PYG_8. Therefore, to build an energy spanner using 3D Yao structures, k needs to be large enough, such as in $FiYG_{56}$ and PYG_{12}. Finally, all above methods based on fixed partition can be performed locally using 1-hop neighbor information and with $O(d)$ time, where d is the number of 1-hop neighbors.

Notice that in [38], the authors also consider the interference effects to adaptively choose the neighbor with the minimum transmit power and adjust the topology. However, such modification will break the connectivity and energy spanner guarantees of 3D Yao structures brought by geometric properties.

3D Yao Structures Based on Flexible Partition

In flexible partition, identical 3D cones with a top angle θ are used to partition the transmission ball and where to define these cones depends on the locations of neighbors around node u (i.e., different nodes may get different partitions). Here θ is an adjustable parameter. Clearly, larger θ leads to lower node out-degree at each node. We use C_{uv} to represent the 3D cone with uv as its axis. In [90, 91], Wang *et al.* proposed three different methods to perform such a partition. However, they showed that the first method (simply applying Yao structure for every 3D cone defined by each neighbor of u and adding the shortest link in each cone) does not bound the node out-degree. See Fig. 7a for such an instance. Let $v_0, v_1, \ldots$ and $w_0, w_1, \ldots$ are all node u's neighbors, their lengths satisfy (1) $\|w_iu\| > \|w_{i+1}u\|$ and $\|w_iu\| < \|v_ju\|$ for any i, j, and (2) $\angle v_iuv_{i+1} = \angle w_iuw_{i+1} = \alpha$. Since w_iu is the shortest link in cone C_{uv_i}, it will be added in the resulting structure so that all w_i are neighbors of u in the final structure. If the angle α is arbitrarily small, the number of nodes w_i could be large, i.e, the node out-degree at u in the resulting structure could be extremely large.

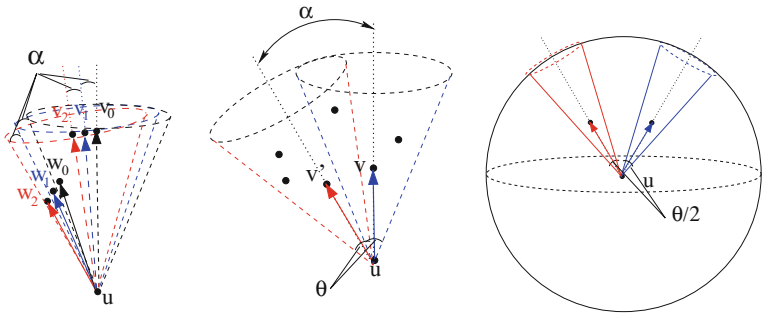


Fig. 7 Illustration of degree bounds for $FlYG_{\theta}$: **a** proof of unbounded degree for the first method in [90, 91]; **b** and **c** proof of bounded degree for the third method in [90, 91]

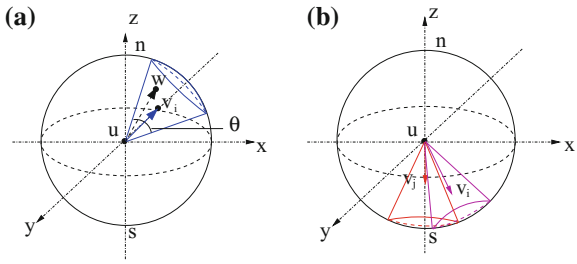


Fig. 8 Definitions of 3D Yao structures with flexible partitions $FlYG_\theta$: **a** 3D cone defined by uv_i and **b** possible overlapping of cones

To fix this problem, the second and third methods by [90, 91] mark the neighbors who have been processed by a 3D cone as *processed* and disable them to be processed by another 3D cone (thus 3D cones are only defined for *unprocessed* links). In the second method, when link uv is in processing, C_{uv} is defined and only the shortest link uw inside the cone is kept, then all links in C_{uv} are marked as *processed*. The third method first orders all links uv in term of link length and then processes them follows an ascending order. Here, we just review their third method and its performance analysis in detail. Initially, all neighbors v_i of node u are *unprocessed* and ordered by the distance to u. The algorithm processes link uv from the shortest link and follows an ascending order. When it processes uv_i, it defines the 3D cone C_{uv_i} which uses uv_i as its axis (as shown in Fig. 8a), adds the link uv_i, and marks all other links in C_{uv_i} as *processed*. We denote the final structure as $FlYG_\theta$ or FlYG when value of θ is clear. Algorithm 1 illustrates the detailed algorithm. The time complexity of this algorithm is $O(d \log d)$ due to the sorting. Notice that the 3D cones in this method are in the same size/shape and can intersect with each other (as in Fig. 8b).

Algorithm 1 Construct 3D Yao Structure $FlYG$ for Node u

Input: all neighbors $N_{UBG}(u)$ of node u in UBG.
Output: neighbors $N_{FlYG}(u)$ of u in the constructed FlYG.

1: Sort all neighbors $v_i \in N_{UBG}(u)$ by its length such that $\|uv_i\| \leq \|uv_{i+1}\|$, where $i = 1$ to $|N_{UBG}(u)|$.
2: Set PROCESSED$(v_i) = 0$ for all neighbor $v_i \in N_{UBG}(u)$.
3: **for** $i = 1$ to $|N_{UBG}(u)|$ **do**
4: **if** PROCESSED$(v_i) = 0$ **then**
5: As shown in Fig. 8a, let C_{uv_i} be the cone using uv_i as the axis and θ as the top angle.
6: Keep v_i as a neighbor of u in FlYG, i.e., add v_i in $N_{FlYG}(u)$.
7: Set PROCESSED$(w) = 1$ for every other neighbor w inside C_{uv_i}.
8: **end if**
9: **end for**
10: **return** $N_{FlYG}(u)$

Now we prove the properties of $FlYG_\theta$ as the following theorem.

Theorem 1. *[90, 91] The node out degree of $FlYG_\theta$ is bounded by $\frac{2}{1-cos(\frac{\theta}{4})}$ and the energy stretch factor of $FlYG_\theta$ is bounded by $\frac{1}{1-(2sin(\frac{\theta}{4}))^\beta}$ when $\theta < 2\pi/3$.*

Proof. In Algorithm 1, after u processes C_{uv}, it marks all links inside C_{uv} as *processed* and those links will never be processed. And each processed cone adds at most one outgoing link in the final structure. Therefore, we only need to prove the number of processed cone is bounded by a constant, then the node out-degree of $FlYG_\theta$ is bounded. Here, we claim that for any two processed cones, the angle α between their axes satisfies $\alpha \geq \theta/2$ as shown in Fig. 7b. Assume there exists any two processed cones C_{uv} and $C_{uv'}$, the angle between their axes $\angle vuv' = \alpha < \theta/2$. Then v' is inside C_{uv} and v is inside $C_{uv'}$. One of v and v' will be processed first, let us assume it is v. Then after adding the shortest link uv in Algorithm 1, u will mark all nodes inside C_{uv} as *processed* including v'. Thus, v' will never be processed which is a contradiction. It is easy to show that the number of processed cones is bounded by a constant. Since $\alpha \geq \theta/2$, the cones with uv and uv' as axes and with $\theta/2$ as top angle cannot intersect each other. Thus, the total number of processed cones is bounded by how many such $\theta/2$ cones can be put into a unit sphere so that they do not intersect with each other. By using a volume argument, this number is bounded by $\frac{4\pi/3}{2\pi(1-cos(\theta/4))/3} = \frac{2}{1-cos(\frac{\theta}{4})}$. See Fig. 7c for reference. This finishes proof of the first part of this theorem about degree bound.

In Algorithm 1, for a link $uw \notin FlYG_\theta$, there must exist a shorter link $uv_i \in FlYG_\theta$ who defined C_{uv_i} where uw is removed. See Fig. 8a. The angle $\angle wuv_i < \theta/2$, since uv_i is the axis. By Lemma 1, the energy stretch factor of $FlYG_\theta$ is $\frac{1}{1-(2sin(\frac{\theta}{4}))^\beta}$ when $\theta < 2\pi/3$. This finishes proof of the second part of this theorem about energy stretch factor.

Notice that if $\theta = \pi/2$, the degree bound is $\frac{2}{1-cos(\frac{\theta}{4})} \approx 26$ which is much smaller than those of $FiYG$s, and if $\theta = \pi/3$, the degree bound is 58.

CBTC-Based Methods

Bahramgiri et al. [11] generalized the *cone-based topology control* (CBTC) protocol [51, 103] from 2D to 3D to preserve connectivity. Basically, each node u increases its transmission power until there is no empty 3D-cone with angle degree α, i.e., there exists at least a node in each 3D-cone of degree α centered at u, if $\alpha \leqslant \frac{2\pi}{3}$. Even though this approach can guarantee connectivity, the gap detection algorithm applied to check the existence of the empty 3D-cone of degree α is very complicated. The time complexity of the gap detection algorithm at a node u is $O(d^3 \log d)$, where d is the node degree of u. Moreover, their method cannot bound node degree, as shown by [60].

Ghosh et al. [31] also presented two CBTC-based approaches for 3D wireless networks. Though the first approach, a heuristic based on 2D orthographic projections, can provide excellent performance in practice, it cannot guarantee connectivity

for sure. In the second approach, a spherical Delaunay triangulation (SDT) is built to determine the existence of empty 3D cones. Although the second approach can guarantee connectivity of the network, the expense to construct the SDT is very high. Similarly, Poduri et al. [70] also used the spherical Delaunay triangulation to find the largest empty 3D cone in order to apply a CBTC-based topology control. The expense of SDT construction makes it inefficient in practice.

2.3.3 3D Yao & Reverse Yao and 3D Symmetric Yao

Even though 3D Yao structures (including FiYG, FlYG, and PYG) can bound the node out-degree, their node in-degree could be as large as $O(n)$ where n is the number of nodes in the networks. Bounded out-degree from 3D Yao structures gives us advantages when applying several routing algorithms on these structures. However, possible unbounded in-degree at some nodes will often cause large overhead or contention at those nodes which may make them exhausted earlier than other nodes. Therefore, it is often imperative to construct a sparse network topology such that both the in-degree and the out-degree are bounded by a constant while it is still energy spanner. Faced up with this challenge, Li et al. [50] proposed two general frameworks to build 3D topologies with bounded node degree (both bounded in-degree and out-degree). Their general frameworks are based on any existing 3D Yao structure (such as FiYG [90, 91], FlYG [90, 91] and PYG [38]). Hereafter, we define a general function 3D-YAO-Structure() which can generate the neighbor set of 3D Yao structure at node u given the current neighbor set of u. The 3D-YAO-Structure() function can be any generation methods of existing Yao-based 3D structures. We use YG to denote the generated 3D Yao structure.

The first set of 3D topologies is *3D Symmetric Yao Graph* (SYG), an undirected graph, which guarantees that the node degree is at most k. It first applies the 3D Yao structure to select the closest node in each 3D cone. An link uv is selected to graph SYG if and only if both u and v are selected to be kept by each other in YG, i.e., $v \in N_{YG}(u)$ and $u \in N_{YG}(v)$. See Fig. 9a and b for illustrations. Algorithm 2 shows the framework. It is clear that only one-hop information is used and total $O(n)$ of messages are used. Thus, the SYG can be built locally and efficiently. Notice that similar idea has been used in 2D networks [57, 83, 96].

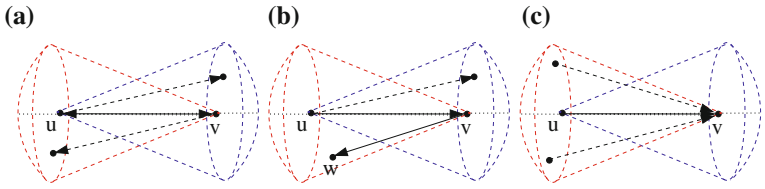


Fig. 9 Illustrations of 3D Yao Structures with bounded degree: **a** and **b** 3D Symmetric Yao Graph; **c** 3D Yao and Reverse Yao Graph. Here, (a) $uv \in$ SYG, (b) $uv \notin$ SYG, (c) $\overrightarrow{uv} \in$ YYG

Algorithm 2 Building 3D Symmetric Yao Graph at Node u

Input: all neighbors $N_{UBG}(u)$ of node u in UBG.
Output: neighbors $N_{SYG}(u)$ of u in the constructed SYG.

1: $N_{YG}(u)$ =3D-YAO-Structure($N_{UBG}(u)$).
2: Broadcast $N_{YG}(u)$ to all neighbors $N_{UBG}(u)$.
3: **for all** node $v \in N_{YG}(u)$ **do**
4: **if** $u \in N_{YG}(v)$ **then**
5: Keep v as a neighbor of u in SYG, i.e., add v in $N_{SYG}(u)$.
6: **end if**
7: **end for**
8: **return** $N_{SYG}(u)$

Algorithm 3 Building 3D Yao and Reverse Yao Graph at Node u

Input: all neighbors $N_{UBG}(u)$ of node u in UBG.
Output: neighbors $N_{YYG}(u)$ of u in the constructed YYG.

1: $N_{YG}(u)$ =3D-YAO-Structure($N_{UBG}(u)$).
2: Broadcast $N_{YG}(u)$ to all neighbors $N_{UBG}(u)$.
3: Let $N^{in}_{YG}(u)$ be the set of u's incoming neighbors, i.e., all node v satisfying $u \in N_{YG}(v)$.
4: $N^{in}_{YYG}(u)$ =3D-YAO-Structure($N^{in}_{YG}(u)$).
5: Broadcast $N^{in}_{YYG}(u)$ to all neighbors $N_{UBG}(u)$.
6: **for all** node $v \in N_{YG}(u)$ **do**
7: **if** $u \in N^{in}_{YYG}(v)$ **then**
8: Keep v as a neighbor of u in YYG, i.e., add v in $N_{YYG}(u)$.
9: **end if**
10: **end for**
11: **return** $N_{YYG}(u)$

The second set of 3D topologies is *3D Yao and Reverse Yao Graph* (YYG), a directed graph, which guarantees that both node in-degree and node out-degree are at most k. The basic idea is to apply reverse 3D Yao structure on YG to bound the node in-degree. Node u chooses a node v from each 3D cone, if there is any, so the incoming link $\overleftarrow{uv}$ in YG has the smallest length among all incoming links from YG in that cone as shown in Fig. 9c. Similar idea has been used for 2D networks by [32, 59, 76]. Algorithm 3 shows the detailed algorithm. 3D YYG can be built locally and efficiently with only 1-hop neighbor information and linear number of messages.

We are now ready to provide some analysis on these 3D structures built by our general frameworks. We will use two basic properties of the underlying 3D Yao structures: (1) the out-degree of 3D YG is bounded by k; and (2) if a link $uv \in UBG$ is not kept in 3D YG, there must exist a shorter link uw kept in 3D YG and $\angle vuw < \theta$. Here θ is the largest angle possible in a 3D cone in FiYG or the half of the top angle of the 3D cone in FlYG.

Theorem 2. *[50] Both 3D SYG and 3D YYG are strongly connected if the original 3D UBG is connected and the angle parameter θ in 3D YG is bounded by $\pi/3$.*

Proof. We first prove the connectivity of 3D SYG, which is equivalent to prove that there is a directed path from u to v in SYG for any two nodes u and v with $||uv|| \leq 1$. We prove this claim by an induction over the distance $||uv||$ between nodes u and v. First, note that the edge between the closest pair of nodes must be kept in SYG. Assume that the claim is true for all links less than $||uv||$. Now we consider nodes u and v. If uv is kept in SYG, the claim is true. If uv is not in SYG, there must a node w inside one of 3D cones at u or v who causes the deletion of uv. Assume w and v are in the same cone of u and $||uw|| < ||uv||$. Because the angle $\angle wuv$ is less than $\theta \leq \frac{\pi}{3}$, we have $||vw|| < ||uv||$. By induction there is a path from u to w and a path from w to v in SYG. Therefore a path from u to v exists in SYG. This finishes the proof for SYG.

We next prove that SYG is a subgraph of YYG, which then can imply the connectivity of 3D YYG automatically. Assume that there exists a link uv in SYG but not in YYG. From the definition of SYG, we know link uv is selected by both u and v in 3D YG. Then if we apply reverse Yao structure on incoming neighbor of 3D YG (as Line 4 in Algorithm 3), uv will also be selected by node v. Thus, uv must be in YYG. This is a contradiction. Therefore, YYG is a supergraph of SYG and fully connected.

Theorem 3. *[50] The node degree of 3D SYG is bounded by k while both node out-degree and in-degree of 3D YYG is bounded by k, where k is the degree bound of underlying 3D YG.*

Proof. This theorem is straightforward from the construction methods of SYG and YYG. Both methods first apply 3D YG. Since each node has at most k 3D cones during this construction, the out-degree is bounded by k. For 3D SYG, a link is kept only if both endpoints keep it in 3D YG. Thus, the node degree of SYG is obviously bounded by k. For 3D YYG, the second round of 3D YG is applied to incoming links, thus the node in-degree is also bounded by k. Notice that in 3D YYG, the out-degree and in-degree neighbors of a node may be different set of nodes.

Theorem 4. *[50] The 3D SYG is not a power spanner of UBG, while 3D YYG is a power spanner of UBG when $\beta \geq 3$ and $\theta < \pi/3$.*

Proof. The first half of this theorem can be directly obtained from a result by Grunewald et al. [32]. They basically show how to construct a counter example of a 2D network in which SYG is not a power spanner. Since the 2D network is a special case of 3D networks, the same counter example works for 3D networks.

The proof of power spanner property of YYG is much challenging, even in 2D. Jia et al. [34] first proved that 2D YYG is a power spanner when $\theta \leq \pi/60$ (i.e., $k \geq 120$). It seems that their proof might be extended to 3D case, however, the node degree bound will be huge (larger than $\frac{4\pi/3}{2\pi(1-\cos(\theta/2))/3} \geq 5836$). Thus it is not very useful in practice. Schindelhauer et al. [79] then proved that 2D YYG is a

power spanner with power spanning ratio $(8c+1)^2 \frac{(2c)^\beta}{1-2^{(2-\beta)}}$ for $\beta > 2$ when $k > 6$. Here $c = \frac{1}{1-2\sin(\pi/k)}$. They proved this by first proving that 2D YYG is a weak c-spanner. In a weak c-spanner, between any two nodes there exists a path which remains within a disk or sphere of radius c-times the Euclidean distance between them. Their proof of weak spanner property of YYG can also be extended to 3D YYG with $\theta < \pi/3$. However, to further extend it to 3D power spanner, it requires $\beta \geq 3$. More specifically, 3D YYG is a power spanner with power spanning ratio $(8c+1)^3 \frac{(2c)^\beta}{1-2^{(3-\beta)}}$ for $\beta > 3$ or $O(c^{12})$ for $\beta = 3$ when $\theta < \pi/3$. Therefore, we can claim that 3D YYG is a power spanner for $\beta \geq 3$ and $\theta < \pi/3$. When $2 \leq \beta < 3$, the power spanner property is still open.

2.3.4 3D Delaunay

Delaunay triangulation [25, 30] is a well-known geomtric spanner. A triangulation of V is a *Delaunay triangulation* (Del), if the circumcircle of each of its triangles do not contain any other vertices of V in its interior in 2D and if the circumsphere of each of its tetrahedrons do not contain any other vertices of V in its interior in 3D (as shown in Fig. 10). In 2D, Del is a planar length spanner which implies it is also an energy spanner. 3D Del keeps the energy spanner property and also has certain benefits for geographic routing (we will discuss them in Sect. 3). However, it is not appropriate to require the construction of Del in the wireless communication environment because of the potentially massive communications it requires. Several published results (localized Delaunay graph (LDel) [55] , partial Delaunay graph [57], and restricted Delaunay graph [29]) were proposed to build 2D Del in a localized way. Similar techniques can be used to build 3D Del for 3D wireless sensor networks, as did in [64].

2.4 Localized 3D Topology Control for Fault Tolerance

In order to be energy efficient, topology control algorithms try to reduce the number of links, and thereby, reduce the redundancy available for tolerating node and link failures. Thus, the topology derived with such algorithms is more vulnerable to node failures or link breakages. However, due to constrained power capacity, hostile deployment environment, and other factors, events like individual node failures are more likely to happen, which might cause network partitions and badly degrade the network performance. Therefore, in order to gain certain degree of redundancy and guarantee the overall performance, fault tolerance becomes an additional and critical requirement for the design of wireless sensor networks. As fault tolerance strongly depends on the network connectivity, topology control for such networks needs to consider both energy efficiency and fault tolerance. Fortunately, most of the 3D structures we introduced so far can be easily extended to support fault tolerance [11, 88, 89]. Next, we briefly review some of them.

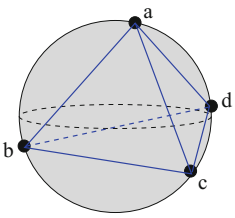


Fig. 10 Definition of 3D Delaunay triangulation: Delaunay tetrahedron *abcd*

2.4.1 3D k-RNG and 3D k-GG

The first two localized structures are based on 3D RNG and 3D GG. The definitions of 3D k-RNG and 3D k-GG are as follows: an edge $uv \in RNG^k$ if and only if the intersection of two balls centered at u and v with radius $\|uv\|$ contains less than k nodes from the set V; an edge $uv \in GG^k$ if and only if the ball with edge uv as a diameter contains less than k nodes of V. See Fig. 11 for illustrations. Clearly, by definitions, 3D RNG and 3D GG are subgraphs of 3D k-RNG and 3D k-GG, respectively. This also implies 3D k-RNG and 3D k-GG preserve the basic connectivity.

Now we prove both 3D k-RNG and 3D k-GG can preserve the k-connectivity, which is much stronger than 1-connectivity.

Theorem 5. *[88, 89] The 3D structures k-RNG and k-GG are k-connected if the UBG G is k-connected, i.e., both k-RNG and k-GG can sustain* $k - 1$ *node faults.*

Proof. We first prove the theorem for 3D k-RNG. Given a set S of $k - 1$ nodes, $S \subset V$, due to the k-connectivity of G, we know that $G - S$ is still connected. To prove k-RNG is k-connected, we prove that k-RNG$-S$ is connected by contradiction. Assume that graph k-RNG$-S$ is not connected, then, there must exist at least a pair of nodes such that there is no path between them. Let the nodes u, v be the pair with the smallest distance to each other , i.e., $\|uv\| \leq \|u'v'\|$ for any pair of nodes u', v' that are not connected. Since $uv \notin$ k-RNG$-S$, $uv \notin$ k-RNG. According to the definition of k-RNG, there should be at least k neighbors w of node u that

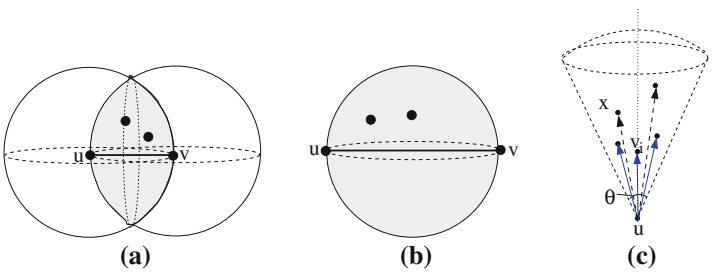


Fig. 11 **a** 3D k-RNG and **b** 3D k-GG: an edge uv is kept if and only if the shaded area has less than k nodes. Here assume $k = 3$, thus uv is kept. **c** 3D k-Yao Graph: k shortest links kept in one cone

satisfy the condition $\|uw\| < \|uv\|$ and $\|wv\| < \|uv\|$. Assume the removed $k-1$ nodes are neighbors of node u, then there is at least one neighbor w left in k-RNG$-S$ with $\|wv\| < \|uv\|$. As nodes u, v is the pair with the smallest distance among those disconnected pairs in k-RNG$-S$, nodes w, v must be connected. Therefore, nodes u, v are also connected via w, which is a contradiction to the assumption. Thus, k-RNG$-S$ is connected, and k-RNG is k-connected. Note that for the case where removed nodes are not all neighbors of u, the proof also holds. The above proof using contradiction can be easily adopted for 3D k-GG too.

Considering the energy efficiency of routes in 3D networks, 3D k-RNG is not an energy spanner for UBG while k-GG is an energy spanner of UBG with stretch factor of one. The latter part can be directly obtained from GG $\subseteq$ k-GG. Next, we consider the situation with at most $k-1$ node failures. Assume that the set of $k-1$ failure nodes is S and UBG(V, E) is the original communication graph. We use UBG* to represent the communication graph without failure nodes and links, i.e., UBG$^* = G(V - S, E - \{uv | u \in S \ \ and \ \ v \in S\})$. Similarly, k-GG* is the graph that removes all failure nodes and links from 3D k-GG. We can prove the following theorem on the power efficiency of 3D k-GG:

Theorem 6. *[88, 89] The structure k-GG* is a power spanner of* UBG* *with spanning ratio bounded by one even with* $k-1$ *node failures* S.

Proof. Basically, we need to prove that every link on the least energy cost paths in UBG* is kept in k-GG*. We prove this by contradiction. Consider any link uv in any least energy cost path in UBG*. Assume that $uv \notin$ k-GG*, thus $uv \notin$ k-GG. By the definition of k-GG, there must be at least k neighbors of node u inside the ball B with edge uv as a diameter. After removing $k-1$ failure nodes S, there must be at least one neighbor left and let us assume it as w. Since w is inside the ball B, $\|uw\|^2 + \|wv\|^2 < \|uv\|^2$. Remember $\beta > 2$, thus $\|uw\|^\beta + \|wv\|^\beta < \|uv\|^\beta$. In other word, the path using uw and wv uses less energy cost than the path using uv. This is a contradiction. It implies that the assumption is wrong, i.e., that edge uv remains in k-GG and k-GG*.

2.4.2 3D k-Yao Structures and CBTC-Based Methods

Both 3D Yao structures and 3D versions of CBTC methods can be extended to support fault tolerance. Wang et al. [88, 89] extended their 3D Yao structures as follows. When the algorithm processes a 3D cone (either in fixed partition or flex partition), it adds the k shortest links in this cone instead of the shortest link only. We denote the final structure as k-YG. Figure 11c shows an example of the algorithm when $k = 3$. Bahramgiri et al. [11] extended their 3D CBTC algorithm to ensure k-connectivity as follows. Each node u increases its transmission power until there is no empty 3D-cone with angle degree α, i.e., there exists at least a node in each 3D-cone of degree α centered at u, if $\alpha \leqslant \frac{2\pi}{3k}$.

Now we prove some properties of k-YG in the following three theorems. Here we use FlYG as the underlying 3D Yao structure. However, similar proofs can be done for other 3D Yao structures.

Theorem 7. *[88, 89] The structure k-YG is k-connected if the original UBG G is k-connected.*

Proof. For simplicity, assume that all $k-1$ fault nodes $v_1, v_2, \ldots, v_{k-1}$ are neighbors of a node u. We show that the remaining graph of k-YG after removing the $k-1$ nodes is still connected. Notice that G is k-connected, thus, the degree of each node is at least k. Additionally, with the $k-1$ fault nodes removed, there is still a path in G to connect any pair of remaining nodes. Assume that the path uses node u and have a link uw, we can prove by induction that there is a path in the remaining graph to connect u and w.

If uw has the smallest distance among all pairs of nodes, according to the construction algorithm uw must be in k-YG. Assume the statement is true for node pair whose distance is the rth shortest. Consider uw with the $(r+1)$th shortest length.

If w is one of the k closest nodes to u in some cone, the link uw remains in the remaining graph. Otherwise, for the cone in which node w resides, there must be other k nodes which are closer to u than w and they are connected with u in k-YG. Since we only have $k-1$ failure nodes, at least one of the links of k-YG in that cone will survive, say link ux. As $\angle xuw < \theta < \frac{\pi}{3}$, xw in triangle xuw is not the longest edge. Thus, $\|xw\| < \|uw\|$, and nodes x and w are connected. Then link uw can be replaced by link ux and a path from x to w by induction. This finishes the proof.

Note that for the case where the nodes removed are not all neighbors of the same node, the induction proof also holds. Induction is based on all pair of nodes.

Theorem 8. *[88, 89] The node out-degree of k-YG is bounded by* $\frac{2k}{1-cos(\frac{\theta}{4})}$.

Proof. Notice that for each processed 3D cone, at most k outgoing links are added in k-YG. Since that the number of processed cones is bounded by $\frac{2}{1-cos(\frac{\theta}{4})}$ as we proved in Theorem 3, the node out-degree of k-YG is bounded by $\frac{2k}{1-cos(\frac{\theta}{4})}$.

For energy efficiency of k-YG, it is obvious that k-YG is an energy spanner of UBG since 1-YG $\subseteq$ k-YG and 1-YG is an energy spanner. Again, we now consider the situation with at most $k-1$ node failures. Assume that S is the set of failure nodes, UBG* and k-YG* are the communication graph without failure nodes/links and the 3D k-YG without failure nodes/links respectively.

Theorem 9. *[88, 89] The structure k-YG* is a power spanner of* UBG* *with spanning ratios bounded by a constant,* $\frac{1}{1-(2sin(\frac{\theta}{4}))^{\beta}}$, *even with* $k-1$ *node failures* S.

Proof. For any link $ux \notin$ k-YG* (also $\notin$ k-YG), there must exist k shorter links in the resulting graph k-YG and k-YG*. Thus, with at most $k-1$ node failures,

there exists at least one shorter link, say uv, and $\angle xuv < \theta/2 < \pi/3$. Therefore, by Lemma 1, k-YG* is still a power spanner of UBG* with spanning ratios bounded by $\frac{1}{1-(2sin(\frac{\theta}{4}))^{\beta}}$.

2.5 Summary

So far, we have introduced several 3D geometric structures which can be constructed locally. Some of them have bounded node degree, some of them are energy spanners of UBG. Table 1 summarizes the properties of existing 3D geometric topologies. Since 3D Del cannot be constructed locally, we list localized Delaunay graph (LDel)[55] instead. Notice that all listed topologies only need 1-hop neighbor information to be constructed, i.e., all construction algorithms are localized algorithms. Thus, when nodes move, the updates of these topologies can be efficiently performed in a local area without any global affects. The time complexity of 3D Yao-based structures is $O(d)$ for fixed partition and $O(d \log d)$ for flexible partition (due to sorting of neighboring links). Here d is the number of 1-hop neighbors. In addition, the energy stretch factor of 3D YYG is $O(1)$ only for $\beta \geq 3$, but still open for $2 \leq \beta < 3$.

In this section, we mainly focus on 3D geometric topologies for energy efficient topology control. But there are also other interesting topics within 3D topology design where geometric techniques can be applied, such as topology design for sensing coverage [35, 101, 102, 112] or connectivity [74] or both coverage and connectivity [5, 8–10, 12, 13, 109]. We believe that geometric approaches will be applied to wider topology design topics in 3D wireless sensor networks far beyond topology control protocols.

3 Geographic Routing for 3D Wireless Sensor Networks

The geometric nature of the multi-hop wireless networks provides a promising idea: localized geographic routing (also called geometric routing, georouting, or position-based routing) [66]. A routing protocol is *localized* if the decision to which node to

Table 1 Properties of proposed and existing 3D geometric topologies

3D structure	Connectivity	Out-degree	In-degree	Power stretch factor	Message	Time
3D LMST	Yes	$O(1)$	$O(1)$	$O(n)$	$O(n)$	$O(d^2)$
3D RNG	Yes	$O(n)$	$O(n)$	$O(n)$	$O(n)$	$O(d)$
3D GG	Yes	$O(n)$	$O(n)$	1	$O(n)$	$O(d)$
3D YG	Yes	$O(1)$	$O(n)$	$O(1)$	$O(n)$	$O(d)$ or $O(d \log d)$
3D SYG	Yes	$O(1)$	$O(1)$	$O(n)$	$O(n)$	$O(d)$ or $O(d \log d)$
3D YYG	Yes	$O(1)$	$O(1)$	$O(1)$	$O(n)$	$O(d)$ or $O(d \log d)$
3D LDel	Yes	$O(n)$	$O(n)$	$O(1)$	$O(n)$	$O(d^2)$

forward a packet is based only on: (1) the information in the header of the packet (including the source and the destination of the packet); (2) the local information gathered by the node from a small neighborhood (i.e., 1-hop neighbors of the node). In this section, we mainly focus on how to design 3D localized geographic routing to achieve packet delivery in large-scale 3D sensor networks.

3.1 Review of 2D Geographic Routing

There have been numerous 2D localized geographic routing protocols [15, 17, 40, 66, 84] proposed in the networking and computational geometry literature. The most common and efficient localized routing is *greedy routing*. In greedy routing, packets are greedily delivered to the neighbor which is the nearest one among the current node and all its neighbors to the destination. Greedy routing has been demonstrated to be very effective in large-scale wireless sensor networks and can be adapted to topology changes dynamically. However, greedy routing fails to deliver the packet when it meets a node which cannot find a neighbor closer to the destination than itself. This problem is called local minimum phenomenon. Such situation often happens at the boundary nodes of topology holes in a wireless sensor network, thus it is also known as routing hole problem of geographic routing. To guarantee the packet delivery after simple greedy heuristic fails at the local minimum, most geographic routing protocols have their own special methods to find a detour path [18]. The most common approach is using face routing as a backup.

Right hand rule is a long-known method for traversing a 2D graph (in analogy to following the right hand wall in a maze). Applying the right-hand rule in planar graphs, *face routing* [40] walk along the faces which are intersected by the line segment from the source to the destination. In each face, it uses the right-hand rule to explore the boundaries. It can guarantee to reach the destination after traversing at most $O(n)$ edges where n is the number of nodes when the underlying network topology is a planar graph. To make face routing more efficient, a natural approach is to combine greedy routing and face routing by using face routing to recover the routing after simple greedy method gets stuck in a local minimum. Many wireless protocols used this approach [17, 37, 41, 43, 84]. For example, both *greedy face routing* (GFG) [17] and *greedy perimeter stateless routing* (GPSR) [37] can guarantee the delivery of the packets by using 2D RNG or GG as the underlying planar routing topology.

Beyond face routing over planar topology, there are also other approaches to build detour paths. For example, in [23], Qing et al. studied how to identify the stuck nodes where greedy forwarding gets stuck in the local minimum and build detour routes around *holes*, which are connected regions of the network with boundaries consisting of all the stuck nodes. In [46], Leong et al. proposed a greedy distributed spanning tree routing (GDSTR) where greedy routing switches to routing on a spanning tree to around the local minimum area. In order to choose a direction on the tree that

is most likely to make progress towards the destination, each node needs to store a summary of the area covered by the subtree below each of its tree neighbors. A nice survey on various detouring methods can be founded at [18]. Recently, there are also several greedy embedding-based methods [39, 78, 108, 110] proposed for 2D sensor networks, which embed the network into certain space such that greedy routing guarantees delivery in the new virtual space.

3.2 3D Greedy Routing

Most classical and widely used localized geographic routing is *greedy routing*, in which a packet is greedily forwarded to the closest node to the destination in order to minimize the average hop count. Greedy routing can be easily extended to 3D case. Actually, several underwater routing protocols [71, 105] for underwater sensor networks are just variations of 3D greedy routing. Figure 12 illustrates the basic idea of 3D greedy routing. Let **t** be the destination node. As shown in Fig. 12a, current node u finds the next relay node v who is the closest to **t** among all neighbors of u. But, it is easy to construct an example (see Fig. 12b) to show that greedy routing will not succeed to reach the destination but fall into a local minimum (at a node without any "better" or "closer" neighbors). This is true for both 2D and 3D networks.

However, to guarantee *packet delivery* of 3D greedy routing is not straightforward and very challenging. Face routing can be used on planar topology to recover from the local minimum of greedy routing and guarantee the delivery in 2D networks, as did in many 2D localized routing protocols [17, 37, 43]. However, there is no planar topology concept any more in 3D networks, thus, face routing cannot be applied directly to help 3D greedy routing get out of local minimum. In the following subsections, we will review several negative or positive results on design of 3D geographic routing with delivery guarantee, and most of them are based on 3D greedy routing.

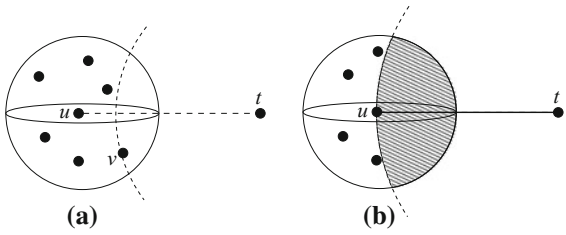


Fig. 12 Illustration of greedy routing in 3D sensor networks: **a** the forwarding neighbor and **b** the local minimum

3.3 None Existence of Deterministic Localized Routing with Delivery Guarantee in 3D Networks

Durocher et al. [22] recently proved that there is *no* deterministic localized routing algorithm for general 3D networks that guarantees the delivery of packets. Next, we briefly review their results and ideas of their proofs.

They first proved that there is a 2-local routing method that guarantees the delivery of packets if the thickness of the 3D network is less than or equal to $\frac{1}{\sqrt{2}}$ times the transmission ranges of nodes. Here a localized routing is k-local if each intermediate node vs routing decision only depends on knowledge of the labels (or positions) of the source and destination nodes and of the k-hop neighborhood of v. This result can be obtained by mapping the UBG into a a d-quasi unit disk graph (d-QUDG) with $d \geq \frac{1}{\sqrt{2}}$, for which a 2-local algorithm with delivery guarantee exists [42]. Here, a d-QUDG is a geometric graph in which any two nodes at distance at most d are always connected, nodes at distance greater than one cannot be connected, and nodes at distance between d and one may or may not be connected.

They then proved that there is no deterministic k-local routing algorithm that guarantees delivery for the class of UBGs contained in thicker slabs, i.e., slabs of thickness $\frac{1}{\sqrt{2}} + \varepsilon$. They proved it by showing that if a k-local routing algorithm were to exist for such UBGs to succeed, then a 1-local algorithm for routing with delivery guarantee would also exist for an arbitrary graph, which is impossible. The detailed steps are as follows. They first showed that any graph G can be translated to a 3D UBG G'. The translation is similar to construct an electronic circuit on three layers with added chains of virtual nodes. They then assumed that there is a k-local routing algorithm A_k with delivery guarantee in 3D UBG. Since the introduction of virtual nodes in UBG G', A_k cannot see more than 1-hop neighbors of the original graph G. The translation from G to G' is strictly local, thus, we can easily simulate 1-local routing algorithm on G. In other words, there also exist a 1-local routing that succeeds for any connected and labelled graph G. Finally, they showed that such 1-local routing does not exist for arbitrary graphs, by constructing a counter example. This disprove the existence of A_k for 3D UBG with a slab of thickness $\frac{1}{\sqrt{2}} + \varepsilon$. This negative result shows that it is very hard to guarantee the packet delivery in 3D geographic routing if only the local information is used.

3.4 3D Routing via Mapping and Projection

Fevens et al. [2, 36] proposed several 3D position-based routing protocols and tried to find a way to still use face routing to get out of the local minimum. Their basic idea is projecting the 3D network to a 2D plane (as shown in Fig. 13a), then applying the face routing in the plane. They called the method *projective face routing* and combine it with the greedy routing. However, as shown in Fig. 13b [36], a planar graph cannot be extracted from the projected graph. It is clear that removing either $v'_3v'_4$ or $v'_1v'_2$ will

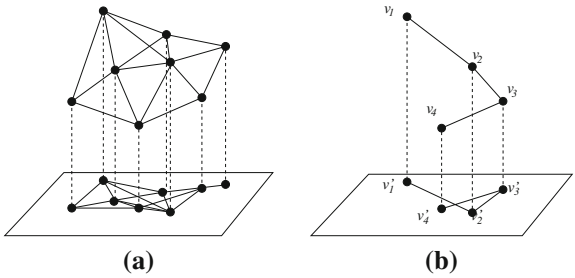


Fig. 13 Simple projection from 3D to 2D does not work: **a** 3D-2D projection and **b** projection causes intersections

break the connectivity. Fevens et al. [2, 36] also proposed *face coordinate routing* which first projects the network onto the xy plane and runs face routing on it. If the face routing fails on the projected graph, it will project the network onto the second plane (the yz plane). If the face routing fails again, the network is projected onto the third plane (the xz plane). However, if the face routing fails on the third plane, this method fails.

In 2D, several greedy embedding algorithms [39, 78, 108, 110] can embed the 2D network into certain space such that greedy routing guarantees delivery in the new virtual space. Unfortunately, none of the greedy embedding algorithms in the literatures can be extended from 2D to 3D general networks.

Even though the projection method cannot solely solve the delivery guarantee problem in 3D geographic routing, it can still be used for parts of combined solutions or even other purposes. For example, Xia et al. [104] recently proposed a deterministic 3D greedy routing which uses volumetric harmonic mapping to map the boundary of a dense unit tetrahedron mesh structure to a sphere, so that greedy routing on the surface of this virtual sphere can guarantee delivery of packets among the boundary nodes. We will review their detailed approach in Sect. 3.9. In [48, 49], Li et al. proposed a 3D circular sailing routing (3D CSR) to balance the traffic load in 3D sensor networks which also projects 3D nodes on a surface of sphere. Under uniform communications, shortest path routing or greedy routing suffers from uneven load distribution in the network, such as crowed center effect where the center nodes have more load than the nodes in the periphery. Aim to balance the load, 3D CSR maps the 3D network onto a 3D or 4D sphere and routes the packets based on the spherical distance on the sphere. Two projection methods to map the nodes in 3D Euclidean space to a sphere are proposed. As shown in Fig. 14, the first projection method maps nodes on the surface of a 3D sphere while the second projection method maps on the surface of a 4D sphere using stereographic projection. Stereographic projection is conformal in any dimension, i.e., it preserves the angles at which curves cross each other and also preserves circles. Therefore, a circle on the sphere is also a circle in the plane (or hyperplane). 3D CSR can calculate the circular distance on the sphere between projected nodes and use it as the routing metric. Since there is no "center" on the sphere surface, the crowed center effect vanishes and the load is balanced.

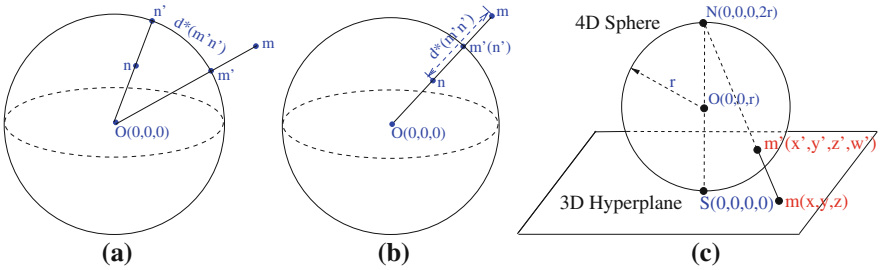


Fig. 14 Projection methods in 3D CSR. **a**, **b** Projection method I: from a node $m(x, y, z)$ in 3D space to a node $m'(x', y', z', \phi)$ on the 3D sphere. There are two cases for the calculation of spherical distance $d^*(m', n')$ as shown in **a** and **b**. **c** Projection method II—stereographic projection: an one-to-one mapping from a node m in a 3D hyperplane to a node m' on a 4D sphere

3.5 Randomized 3D Greedy Routing

Since *no* deterministic localized geographic routing can guarantee the packet delivery [22], randomized algorithms become possible solutions. Now we review two of such approaches based on 3D greedy routing for 3D wireless networks.

In [1], Abdallah et al. proposed a set of randomized geographic routing for 3D networks. First, a randomized algorithm, called *randomized AB3D algorithm*, selects the next hop x randomly from three candidate neighbors n_0, n_1 and n_2 of the current node u. One of the candidate n_0 is the node nearest to the destination $\mathbf{t}$ as 3D greedy selects. The other two candidates n_1 and n_2 are the node chosen by 3D greedy from all neighbors of u above or below the plane defined by n_0, u and $\mathbf{t}$. The probabilities to choose x from these three candidates could be the same or related to the angle $\angle n_i u \mathbf{t}$ or the distance $||n_i \mathbf{t}||$. Then, this randomized AB3D algorithm can be again combined with *face coordinate routing* [2, 36] to form different hybrid 3D routing methods. However, all of these routing methods do not have any performance guarantee.

In [24], Flury and Wattenhofer explored using random walks to escape from the local minimum and proposed a *greedy-random-greedy* (GRG) routing method. The packet is first forwarded greedily until a local minimum is encountered. To resolve the local minimum, a randomized recovery algorithm based on random walk kicks in. Whereas a packet moving around randomly in the network may seem very inefficient and too simplistic, they do proposed several techniques to make random walks more efficient. First, instead of walking on a general graph, they build a sparse virtual graph (called dual graph) which is defined by connecting local spherical structures. By doing so it reduces the search time of random walk significantly. Second, instead of searching the entire network to hit a node which is closer to the destination than current node, the proposed method limits the search space within k-hop. Since the perfect k value is unknown, the recovery algorithm performs an exponential search by limiting the random walk in sequence to 2^i-hop ranges. Last, similar to the face routing in 2D graphs, the proposed method can further restrict the search to nodes delimiting the hole which causes the local minimum and which needs

to be surrounded. Based on the dual graph, the surface can be locally determined by nodes. They also provided a proof that the expected number of hops needed for the random walk method is in the square of the optimal geographic routing algorithm. However, in practice, this randomized method still often leads to high overhead or long delay in 3D networks.

3.6 Critical Transmission Range of 3D Greedy Routing

One way to guarantee the packet delivery for greedy routing in 3D networks is letting all nodes have sufficiently large transmission range to avoid the existence of local minimums. It is clear that this can be achieved when the transmission range is infinite. Assume that V is the set of all wireless nodes in the network and each wireless node has the same transmission range. Let $B(x, r)$ denote the open sphere of radius r centered at x. Let

$$\rho(V) = \max_{\substack{(u,v)\in V^2 \\ u\neq v}} \min_{w\in B(v,\|u-v\|)} \|w-u\|.$$

In the equation, (u, v) is a source-destination pair and $\|u - v\|$ denotes the Euclidean distance between nodes u and v. Since $w \in B(v, \|u - v\|)$, we have $\|w - v\| < \|u - v\|$. It means w is closer to v than u. If the transmission range is not less than $\|w - u\|$, w might be the one to relay packets from u to v. Therefore, for each (u, v), the minimum of $\|w - u\|$ over all nodes on $B(v, \|u - v\|)$ is the transmission range that ensures there is at least one node that can relay packets from u to v, and the maximum of the minimum over all (u, v) pairs guarantees the existence of relay nodes between any source–destination pair. Clearly, if the transmission range is at least $\rho(V)$, packets can be delivered between any source-destination pairs. On the other hand, if the transmission range is less than $\rho(V)$, there must exist some source-destination pair, e.g., the (u, v) that yields the value $\rho(V)$, such that packets can't be delivered. Therefore, $\rho(V)$ is called the *critical transmission range* (CTR) for 3D greedy routing that guarantees the delivery of packets between any source-destination pair of nodes among V.

Recently, Wang et al. [99, 100] studied the critical transmission range for large-scale random 3D networks. Consider a set V of n wireless sensor nodes uniformly distributed in a compact and convex 3D region $\mathbb{D}$ with unit-volume in $\mathbb{R}^3$. By proper scaling, we assume the nodes are represented by a Poisson point process $\mathscr{P}_n$ of density n over a unit-volume cube $\mathbb{D}$. Each node has a uniform transmission range r_n, thus the communication network is modeled by a unit ball graph (UBG) $G(V, r_n)$, where two nodes u and v are connected if and only if their Euclidean distance is at most r_n. The following theorem on CTR $\rho(\mathscr{P}_n)$ of 3D greedy routing in random sensor networks are obtained in [99, 100].

Theorem 10. *[99, 100] Let* $\beta_0 = 3.2$ *and* $n\left(\frac{4}{3}\pi r_n^3\right) = (\beta + o(1)) \ln n$ *for some* $\beta > 0$. *Then, for 3D greedy routing,*

1. *If* $\beta > \beta_0$, *then* $\rho(\mathscr{P}_n) \leq r_n$ *is* asymptotic almost surely.
2. *If* $\beta < \beta_0$, *then* $\rho(\mathscr{P}_n) > r_n$ *is* asymptotic almost surely.

This theorem basically shows that the CTR for 3D greedy routing is *asymptotic almost surely* (a.a.s). at most $\sqrt[3]{\frac{3\beta \ln n}{4\pi n}}$ for any $\beta > \beta_0$ and at least $\sqrt[3]{\frac{3\beta \ln n}{4\pi n}}$ for any $\beta < \beta_0$, where $\beta_0 = 3.2$. This theoretical result answers a fundamental question about how large the transmission range should be set in a 3D sensor networks, such that 3D greedy routing guarantees the delivery of packets between any two nodes with high probability (w.h.p.).

3.7 3D Greedy Routing on Delaunay Triangulation

Recall that in a d-dimensional Euclidean space, a Delaunay triangulation [25, 30] is a triangulation $Del(V)$ such that there is no point in V inside the circum-hypersphere of any d-simplex in $Del(V)$. For example, in 3D space the 3-simplex is a tetrahedron, while in 2D scarce the 2-simplex is a triangle. In [68], Morin proved that 2D greedy routing can guarantee the packet delivery on Delaunay triangulation. This is also true in 3D space, as stated in the following theorem (detailed proof in [99]).

Theorem 11. *[99] The 3D greedy routing can guarantee the packet delivery on any Delaunay triangulation* $Del(V)$.

Delaunay triangulation has been used as routing topology for wireless ad hoc networks [55, 94]. Since building the Delaunay triangulation needs global information and the length of a Delaunay edge could be longer than the maximum transmission range, several methods [55, 57, 94] use local structures to approximate the Delaunay triangulation. This also break the delivery guarantee of 3D greedy routing over them.

Recently, Lam and Qian [44] proposed to use a virtual Delaunay triangulation to aid geographic routing. They called their routing method *multi-hop Delaunay triangulation* (MDT). The key idea is to relax the requirement that every node be able to communicate directly with its neighbor in Delaunay triangulation. In a MDT, the neighbor of a node may not be a physical neighbor. A virtual link represents a multi-hop path between them. When the current node u has a packet with destination $\mathbf{t}$, it forwards to a physical neighbor closest to $\mathbf{t}$ if u is not a local minimum; otherwise the packet is forwarded via a virtual link to a multi-hop Delaunay neighbor closest to $\mathbf{t}$. Due to Theorem 11, MDT can guarantee the packet delivery using a finite number of hops. Simulations also show MDT has low routing stretch from efficient forwarding of packets out of local minimum. In [44] , the authors provided detailed methods to construct and maintain the multi-hop Delaunay triangulation at each node. However, such construction and maintaining are not purely localized. MDT also works for 2D networks or networks with higher dimension.

Liu and Wu [64] also used a Delaunay structure to divide the 3D network into closed subspace and then proposed a greedy-hull-greedy (GHG) routing, which uses

hull routing over the subspace to escape the local minimum and guarantee the delivery. It is a 3D analogue to face routing in 2D. First, a 3D partial unit Delaunay triangulation (PUDT) is constructed to define network hulls (structures corresponding to subspaces) in 3D networks. Here, PUDT construction basically removes intersecting triangles and edges. It can be proven that if there is no intersecting edge and triangle, then there is no overlapping tetrahedra. This is because when two tetrahedra overlap, one of the four triangles on the first tetrahedron must intersect a triangle on the second tetrahedron; moreover, if two triangles intersect, an edge of one of the triangles must intersect the other triangle. Notice that unlike in Delaunay triangulation 3D greedy can encounter a local minimum in PUDT. Once a packet travels to a local-minimum during 3D greedy forwarding in GHG, one of the adjacent hulls of the local-minimum is selected such that the message can recover from the local-minimum by searching the nodes on this hull. GHG selects the hull whose subspace contains the segment connecting the local-minimum and destination, and uses a depth-first-search to travel this hull. Eventually, it can send the message to the node where greedy can be recovered. This local search of possible recover node over the surface of the subspace is very similar to the one used in GRG [24].

3.8 3D Greedy Routing with Spanning Trees

Guarantee delivery can be achieved at the cost of more (non-constant-bounded) storage space. For example, the MDT method [44] need to store and maintain MDT neighbors and paths to them at each node. Zhou et al. [113] also proposed to use hull tree structures (spanning trees) to store possible routes around the void. Such an idea has been used in a 2D geographic routing, GDSTR [46].

In [113], GDSTR is extended to 3D. The new 3D version (GDSTR-3D) uses two hull trees (both spanning trees) for recovery. For each tree, each node stores two 2D convex hulls to aggregate the locations of all descendants in the subtree rooted at the node. The two 2D convex hulls approximate a 3D convex hull at each node to save the storage space.

GDSTR-3D forwards packets greedily as long as it can find a neighbor closer to the destination than the current node. If the packet ends up in a local minimum, the node then attempts to forward the packet to a neighbor that has a neighbor closer to the destination than itself. In other words, GDSTR-3D uses 2-hop 3D greedy routing as the default method. If 2-hop greedy routing still fails, GDSTR-3D switches to forwarding the packet along the edges of a spanning tree which aggregates the location of the nodes in its subtrees using two 2D convex hulls. Since the spanning tree can always reach the destination if the network is connected, GDSTR-3D can always guide the packet to escape from the local minimum and guarantee the delivery.

However, in the worst case routing with the hull tree degrades to depth first search, so the routing path could be long and the storage in a node can be very large. In [104], it has been shown that the storage overhead of GDSTR-3D could be proportional to network size. In addition, some nodes (such as the roots of trees) will be heavily loaded.

3.9 Hybrid 3D Greedy Routing via Unit Tetrahedron Cell Mesh and Volumetric Harmonic Mapping

Xia et al. [104] recently proposed a hybrid 3D greedy routing which uses both a constructed routing structure (unit tetrahedron cell) and a projection method (volumetric harmonic mapping). We now briefly review their geographic routing methods.

First, a unit tetrahedron cell (UTC) mesh structure is constructed from all 3D nodes. A UTC is a tetrahedron formed by four network nodes, which does not intersect with any other tetrahedrons. The union of all UTCs form a mesh structure as shown in Fig. 15b. Notice that UTC mesh is different with 3D Delaunay triangulation, since there maybe nodes inside the circumsphere of a UTC. It is more similar to PUDT [64] in some sense. A simple algorithm to create the UTC mesh has been proposed. However, their method relies on certain assumptions, such as there is no degenerated edges or nodes in the network and any internal hole has been identified, to successfully establish the UTC mesh.

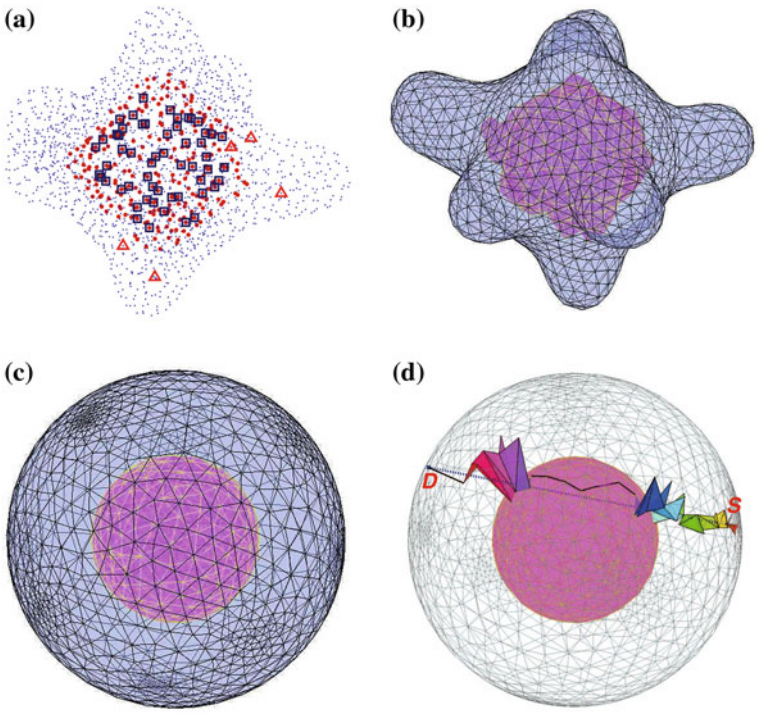


Fig. 15 Illustrations of the hybrid greedy routing method from [104]. **a** A 3D sensor networks where local minimums of node-based greedy are marked as *blue squares* and *red triangles* for boundary nodes and internal nodes respectively. **b** The unit tetrahedron cell mesh structure. **c** Result after volumetric harmonic mapping. **d** The path found by hybrid greedy routing

Second, a face-based greedy routing is proposed to delivery packets within the internal (non-boundary) UTC. The idea is very like the face routing in 2D. The face-based greedy routing will pass a sequence of faces which intersect with the line segment between the source and destination. Each intermediate node can easily calculate the next face by using the information about its neighboring UTCs. It can be proved that such face-based greedy routing does not fail at a non-boundary UTC.

Third, to handle the possible failure of greedy routing at boundaries, the proposed method maps the whole UTC mesh using volumetric harmonic mapping (VHM) under spherical boundary condition (as shown in Fig. 15c) so that the boundary nodes are now on a surface of a sphere. This can guarantee the node-based greedy routing can reach any boundary node successfully. VHM is a one-to-one map that yields virtual coordinates for each node in the entire 3D network to enable global end-to-end greedy routing. In other words, the UTC mesh should remains valid under the virtual coordinates.

Last, a hybrid greedy routing, which alternately uses face-based greedy for internal UTCs and node-based greedy for boundary UTCs, is proposed. Face-based greedy can guarantee the delivery in non-boundary UTCs. When the packet fails at a boundary UTC, node-based greedy is applied to escape the void. Since the boundary has been mapped to a sphere, node-based greedy routing always successes on boundary. When it is possible, it switches back to face-based greedy to route the packet towards the destination. Fig. 15d shows such an example.

Notice that even though a distributed spherical/volumetric harmonic mapping methods are provided in [104], the complexity of such complex procedures still makes the proposed method not very practical. In addition, how to handle multiple inner holes and routing across them is still not clear.

3.10 3D Greedy Routing for Energy Efficiency

Beside the delivery guarantee of packets, the energy efficiency of paths is also very important for 3D wireless sensor networks. Given a routing method $\mathscr{A}$, let $\mathbf{P}_{\mathscr{A}}(s, \mathbf{t})$ be the path found by $\mathscr{A}$ to connect the source node s and the destination node $\mathbf{t}$. A routing method $\mathscr{A}$ is called *energy efficient* if for every pair of nodes s and $\mathbf{t}$, the energy consumption of path $\mathbf{P}_{\mathscr{A}}(s, \mathbf{t})$ is within a constant factor of the least energy-consumption path connecting s and $\mathbf{t}$ in the network. Even a 3D localized routing method can find the route to deliver the packet, it may not guarantee the energy efficiency of the path, i.e., the total power consumed compared with the optimal could be very large in the worst case. Several energy-aware localized 2D/3D routing protocols [47, 67, 71, 80] already took the energy concern into consideration, but none of them can theoretically guarantee the energy-efficiency of their routes. For path energy efficiency, recently, Flury and Wattenhofer [24] proved that *no deterministic localized routing method is energy efficient in 3D networks*. They proved the claim by constructing an example of a 3D network (see Fig. 16) where the path found by

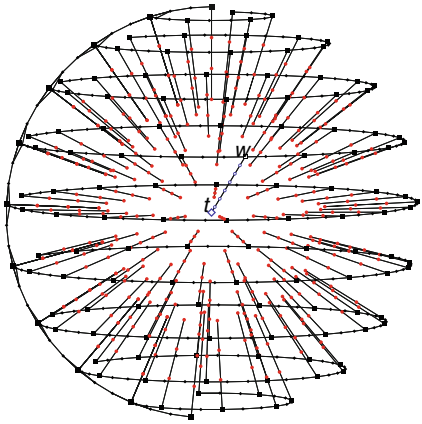


Fig. 16 An example from [24] in which any deterministic localized routing in the worst case needs at least $\Theta(d^3)$ to route a packet from an arbitrary surface node to the center of the sphere

any localized routing protocol to connect two nodes s and t has energy consumption (or hop-count or distance) asymptotically at least $\Theta(d^3)$ in the worst case, where d is the optimum cost.

Since *no* deterministic localized routing protocol is energy-efficient in 3D networks, the simple 3D greedy routing may lead to energy-inefficient paths in the worst case Recently, Wang et al. [97, 99] designed a localized routing method, *energy-efficient restricted greedy* (ERGrd) routing, that is energy-efficient with high probability for random 3D sensor networks. Here a routing method is energy efficient with high probability (w.h.p.) if (1) with high probability, the routing method will find a path successfully; and (2) with high probability, the found path is energy efficient. We now briefly review the idea behind ERGrd routing.

In 3D greedy routing, current node u selects its next hop neighbor based purely on its distance to the destination, i.e., it sends the packet to its neighbor who is closest to the destination. However, such choice might not be the most energy-efficient link locally, and the overall route might not be globally energy-efficient too. Therefore, ERGrd routing uses two concepts *energy mileage* and *restricted region* to refine the choices of forwarding nodes in 3D greedy routing. Given an energy model $\mathbf{e}(x)$ (representing the energy cost for transmit packet over distance x), *energy mileage* is the ratio between the transmission distance and the energy consumption of such transmission, i.e., $\frac{x}{\mathbf{e}(x)}$. Let $\mathbf{r}_0$ be the value such that $\frac{\mathbf{r}_0}{\mathbf{e}(\mathbf{r}_0)} = \max_x \frac{x}{\mathbf{e}(x)}$. We call $\mathbf{r}_0$ as the *maximum energy mileage distance*. We assume that the energy mileage $\frac{x}{\mathbf{e}(x)}$ is an increasing function when $x < \mathbf{r}_0$ and a decreasing function when $x > \mathbf{r}_0$. ERGrd greedily selects the neighbor who can maximize the energy mileage as the forwarding node. In addition, instead of selecting the forwarding node from all neighbors of current node u (a unit ball in 3D as shown in Fig. 17a), ERGrd prefers the forwarding node v inside a smaller restricted region. The region is defined inside a 3D cone with an angle parameter $\alpha < \pi/3$, such that angle $\angle vu\mathbf{t} \leq \alpha$, as shown in Fig. 17b. The use of α (restricting the forwarding direction) is to bound the total distance of the routing path. Then the restricted region is a region inside this 3D cone and

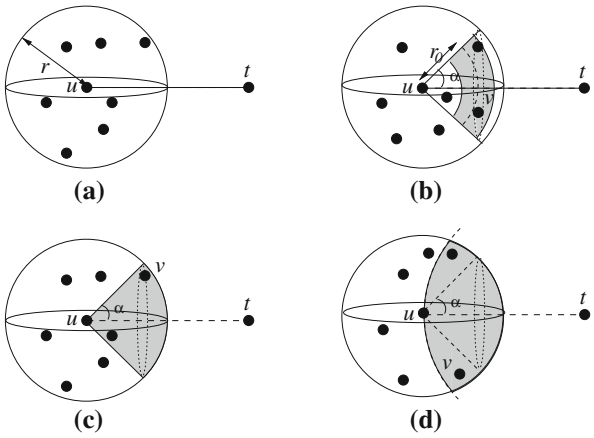


Fig. 17 Illustrations of our 3D routing: **a** all possible forwarding neighbors, **b** best energy mileage forwarding, **c** greedy forwarding inside 3D cone, **d** classical greedy forwarding

near the maximum energy mileage distance $\mathbf{r}_0$, such that every node v inside this area satisfies $\eta_1 \mathbf{r}_0 \leq \|uv\| \leq \eta_2 \mathbf{r}_0$, as shown in Fig. 17b. Here, η_1 and η_2 are two constant parameters. This can help us to prove the energy-efficiency of the route. A detailed algorithm of ERGrd is given in Algorithm 4. There are four parameters used by our method. Three adjustable parameters $0 < \alpha < \frac{\pi}{3}$ and $\eta_1 < 1 < \eta_2$ define the restricted region, while $\mathbf{r}_0$ is the best energy mileage distance based on the energy model. For example, the following setting of these parameters can be used for energy model $\mathbf{e}(x) = x^2 + c$: $\alpha = \frac{\pi}{4}$, $\mathbf{r}_0 = \sqrt{c}$, $\eta_1 = 1/2$ and $\eta_2 = 2$. Notice that if ERGrd fails to find a forwarding node, randomized scheme [24] could also be applied.

Algorithm 4 Energy-Efficient Restricted 3D Greedy Routing (3D ERGrd)

1: **while** node u receives a packet with destination **t do**
2: **if t** is a neighbor of u **then**
3: Node u forwards the packet to **t** directly.
4: **else if** there are neighbors inside the restricted region and $\mathbf{r}_0 < r$ **then**
5: Node u forwards the packet to the neighbor v such that its energy mileage $\frac{\|uv\|}{\mathbf{e}(\|uv\|)}$ is maximum among all neighbors w inside the restricted region, as shown in Fig. 17b.
6: **else if** there are neighbors inside the 3D cone **then**
7: Node u finds the node v inside the 3D cone (Fig. 17c) with the minimum $\|\mathbf{t} - v\|$.
8: **else**
9: Greedy routing (Fig. 17d) is applied, or the packet is simply dropped.
10: **end if**
11: **end while**

The path efficiency of 3D ERGrd is given by the following two theorems [97, 99].

Theorem 12. *[97, 99] When 3D ERGrd routing indeed finds a path $\boldsymbol{P}_{ERGrd}(s,\boldsymbol{t})$ from the source s to the target $\boldsymbol{t}$, the total Euclidean length of the found path is at most $\delta\|\boldsymbol{t}-s\|$ where $\delta=\frac{1}{1-2\sin\frac{\alpha}{2}}$, thus, a constant factor of the optimum.*

Theorem 13. *[97, 99] When 3D ERGrd routing indeed finds a path $\boldsymbol{P}_{ERGrd}(s,\boldsymbol{t})$ from the source s to the target $\boldsymbol{t}$, the total energy consumption of the found path is within a constant factor σ of the optimum. When $\boldsymbol{r}_0 \geq r$, σ depends on α; otherwise, depends on η_1, η_2 and α.*

Notice that 3D ERGrd routing may fail, as all other greedy-based methods do, when an intermediate node cannot find a better neighbor to forward the packet. The CTR of ERGrd routing in random 3D wireless networks can be obtained as the CTR of 3D greedy routing in Sect. 3.6.

Theorem 14. *[97, 99] Let $\beta_0=\frac{2}{1-cos\alpha}$ and $n\left(\frac{4}{3}\pi r_n^3\right)=\beta\ln n$ for some $\beta>0$. Then, for 3D ERGrd routing,*

1. *If $\beta>\beta_0$, then $\rho(\mathscr{P}_n)\leq r_n$ is a.a.s..*
2. *If $\beta<\beta_0$, then $\rho(\mathscr{P}_n)>r_n$ is a.a.s..*

Here, $\beta_0=\frac{4\pi/3}{2\pi(1-cos\alpha)/3}=\frac{2}{1-cos\alpha}$ is the ratio between the volume of a unit ball and the volume of a 3D cone (the forwarding region) inside the ball. Therefore, in summary, by setting the transmission range of each sensor larger than its CTR, 3D ERGrd routing can guarantee the packet delivery and achieve energy-efficient route with high probability in large-scale random sensor networks.

3.11 Summary

In this subsection, we briefly review existing geometric solutions for designing 3D geographic routing to guarantee the packet delivery. These methods are all based on 3D greedy routing but use one or multiple of following techniques to achieve the goal of guaranteed delivery and avoiding the local minimum: storing escaping paths from the local minimum, mapping the network to a virtual space to remove all local minimums, randomly choosing the next hop at a local minimum, enlarging transmission range to eliminate all local minimums, building geometric structures around the surface of a void and traverse around the surface. Table 2 summarizes these 3D routing methods. In summary, to guarantee the delivery in 3D with limited resources is a very challenging task. Beyond the goal of delivery guarantee, there are also other design goals for 3D geographic routings, here we only use energy efficiency as one of such examples. More about geometric approaches for routing in wireless sensor networks can be found in [27].

Table 2 Summary of 3D geographic routing which aims to guarantee the delivery in 3D networks

3D routing	Enlarging TrX range	Projection-based	Randomized	Constructed structure	Hybrid	Localized	Delivery guarantee?
3D greedy						Yes	No
projective face routing [2, 36]		Yes				Yes	No
CFace [2, 36]		Yes (to 2D)				Yes	No
randomized AB3D [1]			Yes			Yes	No
hybrid AB3D-CFace [1]		yes (to 2D)	Yes		Yes	Yes	No
greedy-random-greedy [24]			Yes			Yes	Yes
CTR of 3D greedy [99, 100]	Yes					Yes	Yes (w.h.p.)
MDT [44]				Yes (MDT)			Yes
greedy-hull-greedy [64]				Yes (PUDT)			Yes
GDSTR-3D [113]				Yes (spanning tree)			Yes
VHM based greedy [104]		Yes (VHM)		Yes (UTC)	Yes		Yes
3D ERGrdy [97, 99]	Yes					Yes	Yes (w.h.p.)

4 Conclusion

3D wireless sensor networks have attracted a lot of attention due to their great potential usages in wide range of applications, such as environmental data collection, pollution monitoring, space exploration, disaster prevention, and tactical surveillance. The design of 3D networks is surprisingly more difficult than the design in 2D networks. For example, simply projecting the 3D network into 2D does not work for topology control and geographic routing. Fortunately, the rich geometric properties of 3D wireless sensor network provide new possibilities of applying geometric approaches to address challenging problems (such as localization, topology control, naming, and routing) and provide provable performance guarantee even in a probabilistic and dynamics world. In this chapter, we only focus on the most recent advances in 3D topology control and 3D geographic routing. This is definitely not the whole story. We strongly believe that geometric approaches can be widely used in design and analysis of network protocols for 3D wireless sensor networks. For more topics on geometric approaches in ad hoc and sensor networks (2D or 3D ones), please refer to the following surveys: [27, 28, 61, 87, 95, 98].

Acknowledgments This work was supported in part by the US National Science Foundation (NSF) under Grant No. CNS-0721666, CNS-0915331, and CNS-1050398.

References

1. A. Abdallah, T. Fevens, J. Opatmy, Randomized 3D position-based routing algorithms for ad-hoc networks, in *Proceedings of Annual International Conference on Mobile and Ubiquitous Systems* (2006)
2. A. Abdallah, T. Fevens, J. Opatrnym, Power-aware 3D position-based routing algorithm for ad hoc networks, in *Proceedings of IEEE ICC 2007* (2007)
3. I.F. Akyildiz, D. Pompili, T. Melodia, Underwater acoustic sensor networks: research challenges. Ad Hoc Netw. **3**(3), 257–279 (2005)
4. I.F. Akyildiz, E.P. Stuntebeck, Wireless underground sensor networks: research challenges. Ad Hoc Netw. **4**(6) (2006)
5. S.M.N. Alam, Z.J. Haas, Coverage and connectivity in three-dimensional networks, in *MobiCom '06: Proceedings of the 12th Annual International Conference on Mobile Computing and Networking*, pp. 346–357. ACM Press, New York (2006)
6. J. Allred, A.B. Hasan, S. Panichsakul, W. Pisano, P. Gray, J. Huang, R. Han, D. Lawrence, K. Mohseni, Sensorflock: an airborne wireless sensor network of micro-air vehicles, in *Proceedings of the 5th International Conference on Embedded Networked Sensor Systems, SenSys '07*, pp. 117–129. ACM, New York (2007)
7. K. Alzoubi, X.-Y. Li, Y. Wang, P.-J. Wan, O. Frieder, Geometric spanners for wireless ad hoc networks. IEEE Trans. Parallel Distrib. Processing **14**(4):408–421 (2003). Short version in IEEE ICDCS 2002
8. H.M. Ammari, S.K. Das, Joint k-coverage and hybrid forwarding in duty-cycled three-dimensional wireless sensor networks, in *Proceedings of the 5th IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (IEEE SECON'08)* (2008)

9. H.M. Ammari, S.K. Das, Coverage and connectivity in three-dimensional wireless sensor networks using percolation theory. IEEE Trans. Parallel Distrib. Syst. (IEEE TPDS) **20**(6) (2009)
10. H.M. Ammari, S.K. Das, A study of k-coverage and measures of connectivity in three-dimensional wireless sensor networks. IEEE Trans. Comput. (IEEE TC) **59**(2) (2010)
11. M. Bahramgiri, M.T. Hajiaghayi, V.S. Mirrokni, Fault-tolerant and 3-dimensional distributed topology control algorithms in wireless multi-hop networks, in *Proceedings of the 11th Annual IEEE Internation Conference on Computer Communications and Networks (ICCCN)*, pp. 392–397 (2002)
12. X. Bai, C. Zhang, D. Xuan, W. Jia, Full-coverage and k-connectivity (k=14, 6) three dimensional networks, in *Proceedings of IEEE International Conference on Computer Communications (IEEE INFOCOM)* (2009)
13. X. Bai, C. Zhang, D. Xuan, J. Teng, W. Jia, Low-connectivity and full-coverage three dimensional networks, in *Proceedings of ACM International Symposium on Mobile Ad Hoc Networking and Computing (ACM MobiHoc)* (2009)
14. L. Bao, J. J. Garcia-Luna-Aceves, Topology management in ad hoc networks, in *Proceedings of the 4th ACM international symposium on Mobile ad Hoc Networking & Computing*, pp. 129–140. ACM Press, New York (2003)
15. P. Bose, P. Morin, Online routing in triangulations, in *Proceedings of the 10 th Annual International Symposium on Algorithms and Computation ISAAC* (1999)
16. P. Bose, P. Morin, I. Stojmenovic, J. Urrutia, Routing with guaranteed delivery in ad hoc wireless networks, in *3rd International Workshop on Discrete Algorithms and Methods for Mobile Computing and, Communications* (1999)
17. P. Bose, P. Morin, I. Stojmenovic, J. Urrutia, Routing with guaranteed delivery in ad hoc wireless networks. ACM/Kluwer, Wireless Netw. **7**(6) (2001)
18. D. Chen, P.K. Varshney, A survey of void handling techniques for geographic routing in wireless networks. IEEE Commun. Surv. Tutor. **9**(1–4) (2007)
19. W. Cheng, A. Teymorian, L. Ma, X. Cheng, X. Lu, X. Lu, Underwater localization in sparse 3D acoustic sensor networks, in *Proceedings of the 27th IEEE Conference on Computer Communications (INFOCOM 08)* (2008)
20. A.E. Clementi, G. Huiban, P. Penna, G. Rossi, Y.C. Verhoeven, Some recent theoretical advances and open questions on energy consumption in ad-hoc wireless networks, in *3rd Workshop on Approximation and Randomization Algorithms in Communication, Networks* (2002)
21. X. Dong, M. C. Vuran, Spatio-temporal soil moisture measurement with wireless underground sensor networks, in *Proceedings of 9th IFIP Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)*, pp. 1–8 (2010)
22. S. Durocher, D. Kirkpatrick, L. Narayanan, On routing with guaranteed delivery in three-dimensional ad hoc wireless networks, in *Proceedings of the 9th International Conference on Distributed Computing and Networking (ICDCN)* (2008)
23. Q. Fang, J. Gao, L. Guibas, Locating and bypassing routing holes in sensor networks, in *Proceedings of IEEE INFOCOM '04* (2004)
24. R. Flury, R. Wattenhofer, Randomized 3D geographic routing, in *27th Annual IEEE Conference on Computer Communications (INFOCOM)* Phoenix, USA, April 2008
25. S. Fortune, Voronoi Diagrams and Delaunay Triangulations, in *Computing in Euclidean Geometry*, ed. by F.K. Hwang, D.-Z. Du (World Scientific, Singapore, 1992), pp. 193–233
26. K. Gabriel, R. Sokal, A new statistical approach to geographic variation analysis. Syst. Biol. **18**, 259–278 (1969)
27. J. Gao, *Geometric routing in wireless sensor networks* Book Chapter of Guide to Wireless Sensor Networks (Springer, Berlin, 2009)
28. J. Gao, L.J. Guibas, Geometric algorithms for sensor networks. Phil. Trans. R. Soc. A **307**, 2012 (1958)
29. J. Gao, L.J. Guibas, J. Hershburger, L. Zhang, A. Zhu, Geometric spanner for routing in mobile networks, in *Proceedings of the 2nd ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 01)* (2001)

30. P.-L. George, H. Borouchaki, *Delaunay Triangulations and Meshing* (HERMES, Paris, 1998)
31. A. Ghosh, Y. Wang, B. Krishnamachari, Efficient distributed topology control in 3-dimensional wireless networks, in *4th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and, Networks, SECON'07*, pp. 91–100, June 2007
32. M. Grünewald, T. Lukovszki, C. Schindelhauer, K. Volbert, Distributed maintenance of resource efficient wireless network topologies, in *Proceedings of the 8th European Conference on Parallel Computing (Euro-Par'02)* (2002)
33. X. Hong, M. Gerla, R. Bagrodia, T. Kwon, P. Estabrook, G. Pei, The Mars sensor network: efficient, energy aware communications, in *IEEE Military Communications Conference (MILCOM 2001)* (2001)
34. L. Jia, R. Rajaraman, C. Scheideler, On local algorithms for topology control and routing in ad hoc networks, in *Proceedings of the 15th Annual ACM Symposium on Parallel Algorithms and Architectures* (2003)
35. M. Jin, G. Rong, H. Wu, L. Shuai, X. Guo, Optimal surface deployment problem in wireless sensor networks, in *Proceedings of the 31st IEEE INFOCOM* (2012)
36. G. Kao, T. Fevens, J. Opatrny, Position-based routing on 3D geometric graphs in mobile ad hoc networks, in *CCCG 2005* (2005)
37. B. Karp, H. Kung, GPSR: Greedy perimeter stateless routing for wireless networks, in *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)* (2000)
38. J. Kim, J. Shin, Y. Kwon, Adaptive 3-dimensional topology control for wireless ad-hoc sensor networks, in *IEICE Transactions*, pp. 2901–2911 (2010)
39. R. Kleinberg, Geographic routing using hyperbolic space, in *Proceedings of the 26th Conference of the IEEE Communications Society (INFOCOM07)* (2007)
40. E. Kranakis, H. Singh, J. Urrutia, Compass routing on geometric networks, in *Proceedings of 11th Canadian Conference on Computational Geometry*, pp. 51–54 (1999)
41. F. Kuhn, R. Wattenhofer, Y. Zhang, A. Zollinger, Geometric ad-hoc routing: of theory and practice, in *Proceedings of the ACM International Symposium on the Principles of Distributed Computing (PODC)* (2003)
42. F. Kuhn, R. Wattenhofer, A. Zollinger, Ad-hoc networks beyond unit disk graphs, in *1st ACM Joint Workshop on Foundations of Mobile Computing (DIALM-POMC)*, San Diego, California, USA, September 2003
43. F. Kuhn, R. Wattenhofer, A. Zollinger, Worst-case optimal and average-case efficient geometric ad-hoc routing, in *Proceedings of the 4th ACM International Symposium on Mobile Ad-Hoc Networking and Computing (MobiHoc)* (2003)
44. S.S. Lam, C. Qian, Geographic routing in d-dimensional spaces with guaranteed delivery and low stretch, in *Proceedings of the ACM SIGMETRICS joint international conference on Measurement and Modeling of Computer Systems, SIGMETRICS '11*, pp. 257–268. ACM, New York (2011)
45. S. Lee, B. Bhattacharjee, S. Banerjee, Efficient geographic routing in multihop wireless networks, in *MobiHoc '05: Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and Computing*, pp. 230–241. ACM Press, New York (2005)
46. B. Leong, B. Liskov, R. Morris, Geographic routing without planarization, in *Proceedings of the 3rd Conference on Networked Systems Design & Implementation* (2006)
47. C.-P. Li, W.-J. Hsu, B. Krishnamachari, A. Helmy, A local metric for geographic routing with power control in wireless networks, in *Proceedings of IEEE SECON* (2005)
48. F. Li, S. Chen, Y. Wang, Load balancing routing with bounded stretch, in *EURASIP Journal on Wireless Communications and Networking* (2010)
49. F. Li, S. Chen, Y. Wang, J. Chen, Load balancing routing in three dimensional wireless networks, in *2008 IEEE International Conference on, Communications (ICC2008)* (2008)
50. F. Li, Z. Chen, Y. Wang, Localized geometric topologies with bounded node degree for three dimensional wireless sensor networks. EURASIP J. Wireless Commun. Netw. **1**, 2012 (2012)

51. L. Li, J.Y. Halpern, P. Bahl, Y.-M. Wang, R. Wattenhofer, Analysis of a cone-based distributed topology control algorithms for wireless multi-hop networks, in *ACM Symposium on Principle of Distributed Computing (PODC)* (2001)
52. M. Li, Y. Liu, Underground coal mine monitoring with wireless sensor networks. ACM Trans. Sen. Netw. **5**(2):10:1–10:29 (2009)
53. N. Li, J.C. Hou, FLSS: a fault-tolerant topology control algorithm for wireless networks, in *MobiCom '04: Proceedings of the 10th Annual International Conference on Mobile Computing and Networking*, pp. 275–286. ACM Press, New York (2004)
54. N. Li, J. C. Hou, L. Sha, Design and analysis of a MST-based topology control algorithm, in *Proceedings of IEEE INFOCOM 2003* (2003)
55. X.-Y. Li, G. Calinescu, P.-J. Wan, Y. Wang, Localized delaunay triangulation with application in wireless ad hoc networks. IEEE Trans. Parallel Distrib. Processing **14**(10), 1035–1047 (2003)
56. X.-Y. Li, W.-Z. Song, W. Wang, A unified energyefficient topology for unicast and broadcast, in *11th ACM Annual International Conference on Mobile Computing and Networking (MobiCom 2005)* (2005)
57. X.-Y. Li, I. Stojmenovic, Y. Wang, Partial delaunay triangulation and degree limited localized Bluetooth multihop scatternet formation. IEEE Trans. Parallel Distrib. Syst. **15**(4), 350–361 (2004)
58. X.-Y. Li, P.-J. Wan, Y. Wang, Power efficient and sparse spanner for wireless ad hoc networks, in *IEEE International Conference Computer Communications and Networks (ICCCN01)*, pp. 564–567 (2001)
59. X.-Y. Li, P.-J. Wan, Y. Wang, O. Frieder, Sparse power efficient topology for wireless networks, in *IEEE Hawaii International Conference on System Sciences (HICSS)* (2002)
60. X.-Y. Li, P.-J. Wan, Y. Wang, C.-W. Yi, O. Frieder, Fault tolerant deployment and topology control in wireless ad hoc networks. Wiley J. Wireless Commun. Mob. Comput. **4**(1), 109–125 (2004)
61. X.-Y. Li, Y. Wang, in *Wireless Sensor Networks and Computational Geometry*, ed. by M. Ilyas et al. Book Chapter of Handbook of Sensor Networks (CRC Press Boca Raton, 2004). ISBN: 0-8493-1968-4.
62. X.-Y. Li, Y. Wang, W.-Z. Song, Applications of k-local MST for topology control and broadcasting in wireless ad hoc networks. IEEE Trans. Parallel Distrib. Syst. **15**(12), 1057–1069 (2004)
63. B. Liang, Z.J. Haas, Virtual backbone generation and maintenance in ad hoc network mobility management. Proceedings of the Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies INFOCOM **3**, 1293–1302 (2000)
64. C. Liu, J. Wu, Efficient geometric routing in three dimensional ad hoc networks, in *Proceedings of 27th Annual Joint Conference of IEEE Communication and Computer Society (INFOCOM) (Mini-conference)* (2009)
65. L. Lloyd, R. Liu, M.V. Marathe, R. Ramanathan, S.S. Ravi, Algorithmic aspects of topology control problems for ad hoc networks, in *ACM MOBIHOC* (2002)
66. M. Mauve, J. Widmer, H. Harenstein, A survey on position-based routing in mobile ad hoc networks. IEEE Netw. Mag. **15**(6), 30–39 (2001)
67. T. Melodia, D. Pompili, I.F. Akyildiz, Optimal local topology knowledge for energy efficient geographical routing in sensor networks, in *IEEE INFOCOM* (2004)
68. P. Morin, Online routing in Geometric Graphs. PhD thesis, Carleton University School of Computer Science, 2001
69. N. Nikaein, C. Bonnet, Topology management for improving routing and network performances in mobile ad hoc networks. Mob. Netw. Appl. **9**(6), 583–594 (2004)
70. S. Poduri, S. Pattem, B. Krishnamachari, G.S. Sukhatme, Using local geometry for tunable topology control in sensor networks, in *IEEE Transactions on Mobile Computing*, pp. 218–230 (2009)
71. D. Pompili, T. Melodia, Three-dimensional routing in underwater acoustic sensor networks, in *Proceedings of ACM PE-WASUN 2005*, Montreal, Canada, October 2005

72. R. Rajaraman, Topology control and routing in ad hoc networks: a survey. SIGACT News **33**, 60–73 (2002)
73. R. Ramanathan, R. Hain, Topology control of multihop wireless networks using transmit power adjustment. IEEE INFOCOM **2**, 404–413 (2000)
74. V. Ravelomanana, Extremal properties of three-dimensional sensor networks with applications. IEEE Trans. Mob. Comput. **3**(3), 246–257 (2004)
75. R.S. Roberts, A cooperative uav-based communications backbone for sensor networks. White Paper UCRL-ID-145787, Lawrence Livermore National Laboratory (2001)
76. S. Rührup, C. Schindelhauer, K. Volbert, M. Grünewald, Performance of distributed algorithms for topology control in wireless networks, in *Proceedings of the 17th International Parallel and Distributed Processing Symposium (IPDPS 2003)* (2003)
77. P. Santi, *Topology Control in Wireless Ad Hoc and Sensor Networks* (Wiley, New York, 2005)
78. R. Sarkar, X. Yin, J. Gao, F. Luo, X.D. Gu, Greedy routing with guaranteed delivery using Ricci flows, in *Proceedings of the 8th International Symposium on Information Processing in Sensor Networks (IPSN'09)*, pp. 121–132, April 2009
79. C. Schindelhauer, K. Volbert, M. Ziegler, Geometric spanners with applications in wireless networks. Comput. Geom. Theory Appl. **36** (2005)
80. K. Seada, M. Zuniga, A. Helmy, B. Krishnamachari, Energy-efficient forwarding strategies for geographic routing in lossy wireless sensor networks, in *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems, SenSys '04*, pp. 108–121. ACM, New York (2004)
81. M. Seddigh, J.S. Gonzalez, I. Stojmenovic, RNG and internal node based broadcasting algorithms for wireless one-to-one networks. ACM Mob. Comput. Commun. Rev. **5**(2), 37–44 (2002)
82. W.-Z. Song, Y. Wang, X.-Y. Li, Localized algorithms for energy efficient topology in wireless ad hoc networks. ACM/Springer Mob. Netw. Appl (MONET) **10**(6), 911–923 (2005)
83. I. Stojmenovic, Dominating set based bluetooth scatternet formation with localized maintenance, in *Proceedings of the 16th International Parallel and Distributed Processing Symposium, IPDPS '02*, pp. 122–129. IEEE Computer Society, Washington, DC (2002)
84. I. Stojmenovic, X. Lin, Loop-free hybrid single-path/flooding routing algorithms with guaranteed delivery for wireless networks. IEEE Trans. Parallel Distrib. Syst. **12**(10) (2001)
85. G.T. Toussaint, The relative neighborhood graph of a finite planar set. Pattern Recogn. **12**(4), 261–268 (1980)
86. P. Vincent, M. Tummala, J. McEachen, A new method for distributing power usage across a sensor network, in *IEEE SECON 2006* (2006)
87. Y. Wang, in *Topology Control for Wireless Sensor Networks*, ed. by Y. Li, M. Thai, W. Wu. Book Chapter of Wireless Sensor Networks and Applications (Springer, Berlin, 2007). ISBN: 978-0-387-49591-0
88. Y. Wang, L. Cao, T.A. Dahlberg, Efficient fault tolerant topology control for three-dimensional wireless networks, in *17th IEEE International Conference on Computer Communications and Networks (ICCCN 2008)* (2008)
89. Y. Wang, L. Cao, T.A. Dahlberg, F. Li, X. Shi, Self-organizing fault tolerant topology control in large-scale three-dimensional wireless networks. ACM Trans. Auton. Adapt. Syst. (TAAS) **4**(3):19:1–19:21 (2009)
90. Y. Wang, F. Li, T. Dahlberg, Power efficient 3-dimensional topology control for ad hoc and sensor networks, in *IEEE Global Telecommunications Conference (GlobeCom 2006)* (2006)
91. Y. Wang, F. Li, T.A. Dahlberg, Energy-efficient topology control for 3-dimensional sensor networks. Int. J. Sens. Netw. (IJSNet) **4**(1/2), 68–78 (2008)
92. Y. Wang, X.-Y. Li, Localized construction of bounded degree and planar spanner for wireless ad hoc networks. ACM/Springer Mob. Netw. Appl (MONET) **11**(2), 161–175 (2006)
93. Y. Wang, X.-Y. Li, Minimum power assignment in wireless ad hoc networks with spanner property. J. Combin. Optim. **11**(1), 99–112 (2006)
94. Y. Wang, X.-Y. Li, Efficient delaunay-based localized routing for wireless sensor networks. Wiley Int. J. Commun. Syst. **20**(7), 767–789 (2007)

95. Y. Wang, X.-Y. Li, in Geometrical Spanner for Wireless Ad Hoc Networks, ed. by T.F. Gonzalez. Book Chapter of Handbook on Approximation Algorithms and Metaheuristics (Chapman & Hall/CRC, New York, 2007)
96. Y. Wang, X.-Y. Li, O. Frieder, Distributed spanner with bounded degree for wireless networks. Int. J. Found. Comput. Sci. **14**(2), 183–200 (2003)
97. Y. Wang, X.-Y. Li, W.-Z. Song, M. Huang, T.A. Dahlberg, Energy-efficient localized routing in random multihop wireless networks. IEEE Trans. Parallel Distrib. Syst. (TPDS) **22**(8), 1249–1257 (2011)
98. Y. Wang, Y. Liu, Z. Guo, Three-dimensional ocean sensor networks: a survey. J. Ocean Univ. China **11**(4), 436–450 (2012)
99. Y. Wang, C.-W. Yi, M. Huang, F. Li, Three dimensional greedy routing in large-scale random wireless sensor networks. Ad Hoc Netw. J. (to appear)
100. Y. Wang, C.-W. Yi, F. Li, Delivery guarantee of greedy routing in three dimensional wireless networks, in *International Conference on Wireless Algorithms, Systems and Applications (WASA08)* (2008)
101. M. Watfa, S. Commuri, The 3-dimensional wireless sensor network coverage problem, in *Proceedings of the 2006 IEEE International Conference on Networking, Sensing and Control (ICNSC '06)* (2006)
102. M. Watfa, S. Commuri, Optimal 3-dimensional sensor deployment strategy, in *3rd IEEE Consumer Communications and Networking Conference (CCNC 2006)* (2006)
103. R. Wattenhofer, L. Li, P. Bahl, Y.-M. Wang, Distributed topology control for power efficient operation in multihop wireless ad hoc networks, in IEEE INFOCOM'01 (2001)
104. S. Xia, X. Yin, H. Wu, M. Jin, X.D. Gu, Deterministic greedy routing with guaranteed delivery in 3D wireless sensor networks, in *Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, pp. 1–10 (2011)
105. P. Xie, J.-H. Cui, L. Lao, VBF: vector-based forwarding protocol for underwater sensor networks, in *Proceedings of IFIP Networking'06* (2006)
106. N. Xu, S. Rangwala, K.K. Chintalapudi, D. Ganesan, A. Broad, R. Govindan, D. Estrin, A wireless sensor network for structural monitoring, in Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems, SenSys '04, pp. 13–24. ACM, New York (2004)
107. A.C.-C. Yao, On constructing minimum spanning trees in k-dimensional spaces and related problems. SIAM J. Comput. **11**, 721–736 (1982)
108. W. Zeng, R. Sarkar, F. Luo, X. D. Gu, J. Gao, Resilient routing for sensor networks using hyperbolic embedding of universal covering space, in *Proceedings of the 29th Annual IEEE Conference on Computer Communications (INFOCOM10)* (2010)
109. C. Zhang, X. Bai, J. Teng, D. Xuan, W. Jia, Constructing low-connectivity and full-coverage three dimensional sensor networks. IEEE J. Sel. Areas Commun. (JSAC), **28**(7) (2010)
110. F. Zhang, H. Li, A.A. Jiang, J. Chen, P. Luo, Face tracing based geographic routing in nonplanar wireless networks, in *Proceedings of the 26th Conference of the IEEE Communications Society (INFOCOM07)* (2007)
111. Y. Zhang, L. Cheng, A distributed protocol for multi-hop underwater robot positioning, in *Proceedings of IEEE International Conference on Robotics and Biomimetics* (2004)
112. M. Zhao, J. Lei, M. Wu, Y. Liu, W. Shu, Surface coverage in wireless sensor networks, in *Proceedings of the 28th IEEE INFOCOM* (2009)
113. J. Zhou, Y. Chen, B. Leong, P.S. Sundaramoorthy, Practical 3D geographic routing for wireless sensor networks, in *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems, SenSys '10*, pp. 337–350. ACM, New York (2010)
114. Z. Zhou, J.-H. Cui, S. Zhou, Localization for large-scale underwater sensor networks, in *Proceedings of IFIP Networking'07* (2007)
115. Z. Zhou, S. Das, H. Gupta, Fault tolerant connected sensor cover with variable sensing and transmission ranges, in *Second Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks*, pp. 594–604, September 2005

116. Y. Zhu, M. Huang, S. Chen, Y. Wang, Cooperative energy spanners: Energy-efficient topology control in cooperative ad hoc networks, in *IEEE 30th Conference on Computer Communications (INFOCOM11)*, Mini-conference (2011)
117. Y. Zhu, M. Huang, S. Chen, Y. Wang, Energy-efficient topology control in cooperative ad hoc networks. IEEE Trans. Parallel Distrib. Syst. (TPDS) **23**(8), 1480–1491 (2011)

Chapter 11
Routing in Three-Dimensional Wireless Sensor Networks

Anne Paule Yao and Habib M. Ammari

Abstract Advances in wireless sensor networks (WSNs) technology have been undergoing a revolution that promises a significant impact on society. Most existing wire-less systems and protocols are based on two-dimensional design, where all wireless nodes are distributed in a two-dimensional (2D) plane. However, 2D assumption may no longer be valid if a wireless network is deployed in space, atmosphere, or ocean, where nodes of a network are distributed over a three-dimensional (3D) space and the differences in the third dimension are too large to be ignored. In fact, recent interest in wireless sensor networks hints at the strong need to design 3D wireless networks. The characteristics of 3D wireless sensor networks require more effective methods to ensure routing and data dissemination protocols in these networks. In this chapter, we present a survey of the state-of-the-art routing techniques in 3D WSNs.

1 Introduction

Three-dimensional (3D) wireless sensor networks (WSNs) have recently emerged as a premier research topic and have attracted a lot of attention due to their great potential usages in both commercial and civilian applications, such as environmental data collection, pollution monitoring, space exploration, disaster prevention, and tactical surveillance. Several current researches in 3D networks focus on coverage [1–4], connectivity [4, 5], topology control [6, 7], and routing issues [8–14] and protocols [15–20]. Since it is well known that the sensor nodes have a limited transmission range, and that these sensors have a limited processing and storage capabilities as well as scarce energy resources, specific routing protocols are designed for these networks. They are responsible for maintaining the routes in the

A. P. Yao · H. M. Ammari (✉)
College of Engineering, Department of Computer and Information Science,
WiSeMAN Research Lab, University of Michigan-Dearborn, Dearborn, MI 48128, USA
e-mail: hammari@umd.umich.edu

H. M. Ammari (ed.), *The Art of Wireless Sensor Networks*,
Signals and Communication Technology, DOI: 10.1007/978-3-642-40066-7_11,

network and have to ensure reliable multi-hop communication under these conditions. Most existing wireless sensor systems and protocols are based on two-dimensional (2D) design, where all wireless sensor nodes are distributed in a two-dimensional plane. This assumption is somewhat justified for applications where sensor nodes are deployed on earth surface and where the height of the network is smaller than transmission radius of a node. However, 2D assumption may no longer be valid if a wireless sensor network is deployed in space, atmosphere, or ocean, where nodes of a network are distributed over a 3D space and the difference in the third dimension is too large to be ignored. In fact, recent interests in under-water sensor networks [21] or space sensor networks [22] hints at the strong need to design 3D wireless networks and 3D settings reflect more accurate network design for real-world applications. For example, a network deployed on the trees of different heights in a forest, in a building with multiple floors, or underwater [21], requires design in 3D rather than 2D space. Thus, increasingly today, some efforts have been devoted to the design of routing and data dissemination protocols for 3D sensing applications. This chapter surveys recent routing protocols for 3D Wireless Sensor Networks and presents a classification for the various approaches pursued. We first outline the network characteristics and design challenges for routing protocols in 3D WSNs followed by a presentation and definition of terms and concepts used in 3D WSNs. Then, we focus on a comprehensive survey of different routing techniques for 3D WSNs. And, in the following part, we make a comparison between the routing protocols. The chapter concludes with possible future research areas.

1.1 Motivation for 3D WSNs Routing Protocols

Today, wireless communication is essential since space and water are used as communication medium. Moreover, in order to capture any information in any place, a huge number of nodes which may rise up to thousands are needed. Theses nodes should have some characteristics to properly feature any type of environment and application. Thus, the nodes should be small, with sensing capabilities for environment monitoring, low power, low bit rate, low cost, and autonomous. The applications based on WSNs are then multiple and very varied: Target Tracking, military and security issues, environmental monitoring, health, home, space exploration, chemical processing, disaster relief, etc.

Moreover, to be able to sense every kind of information, nodes are now deployed in the three dimensions (forests, buildings, oceans, etc.). 3D space is essential in most of these applications. Then, 3D wireless sensor networks are best suitable to meet the requirements. Consequently, 3D WSN routing protocols which help to implement these applications by finding the best way to route the data such that network requirements are meet, have an important role to play. Routing protocols define the mode operation of the network.

2 Network Characteristics and Routing Issues

The characteristics of sensor networks and application requirements have a significant impact on the network design objectives and routing protocols in term of network capabilities and network performance [23].

2.1 Network Design in 3D WSNs

Wireless sensor networks in three dimensions have several applications from military to civilian through those environmental applications such as surveillance, target tracking, monitoring, and disaster prevention. These wireless systems are designed according to sensor networks specific characteristics and they are also based on the applications objectives. This part focuses on the main characteristics of 3D WSNs and on the design objectives of these 3D networks.

2.1.1 Network Characteristics of 3D WSNs

Wireless sensor networks whether in two dimensions or three dimensions have certain distinctive features of wireless sensor networks. Although sensor networks in 2D and 3D in general have the same characteristics, these characteristics are accentuated when the network have to be designed for high dimensions (dimension ≥ 2) [24]. As a reminder, sensor nodes can be imagined as small computers, extremely basic in terms of their interfaces and their components. They usually consist of a processing unit with limited computational power and limited memory, sensors, a communication device (usually radio transceivers or alternatively optical), and a power source usually in the form of a battery [25]. The main characteristics of these networks include [26]:

Dense sensor node deployment: Sensor nodes are usually densely deployed. In 3D space, in order to cover the entire considered space, node deployment can be very large, several orders of magnitude higher than that in 2D WSNs. Node deployment has two types: random or deterministic (e.g., grid-based deployment) [27].

Power consumption constraints: For nodes using batteries, these ones determine lifetime of each sensor and of the entire network. There are also sensors nodes that use energy harvesting and that are subject to the same restricted energy constraint.

Severe computation and storage constraints: Sensor nodes in 3D have to deal with much more processes—due to the high number of nodes and the dimension of the workspace (dimension ≥ 3)—despite their highly limited computation, and storage capabilities. Well in 3D, computation and storage constraints are emphasized.

Ability to cope with failures (of nodes and communication): The sensor network can be subjected to communication problems and faulty or inaccuracy data both related to the quality of nodes and links between them. Sensor nodes are not always reliable since they are prone to physical damage or failures due to its deployment environment (air, ocean or even building) and maintenance. Moreover, even though

nodes can be reliable, communications links between them may fail because of nodes deployment, some interferences or blockages. These failures are different from the losses on communication channels such as path loss which remains intrinsic to the wireless channel.

Asymmetry in the communication: Due to obstacles between two nodes, the communication between them can be affected such that the volume of transmitted data is much greater in one direction than the other or the speed of transmitted data is much greater in one direction than the other or even the communication is possible in only one direction. The sensor network copes with asymmetry in the communication.

Nodes mobility: Sensor nodes are able or free to move and change place in the network.

Frequent topology change: The mobility leads to a frequent topology change characteristic of these networks. This also happens when a node joins, leaves the network, or fails.

Self-configurable: Sensor nodes are usually randomly deployed and autonomously configure themselves into a communication network.

Ability to withstand harsh environmental conditions: Since 3D sensor nodes are deployed everywhere in the atmosphere, and underwater, sensor nodes should be able to resist to very hard natural conditions.

Ease of use: Sensor nodes are simple computing systems very easy to deploy and to use.

Data redundancy: Sensor nodes collaborate to accomplish sensing task. The more, networks are dense, the more the data sensed by multiple sensor nodes typically have a certain level of correlation or redundancy.

Application specific: A sensor network is usually designed and deployed for a specific application. The design requirements of a sensor network change with its application.

Pattern of node deployment: In most sensor network applications, the data sensed by sensor nodes flow from multiple source sensor nodes to a particular sink, exhibiting a many-to-one traffic pattern.

Node type (Homogeneous/Heterogeneous): The set of nodes that are selected for a sensor network can be either a homogeneous or heterogeneous group of nodes. A homogeneous group is a group in which all of the nodes have the same capabilities. A heterogeneous group is one in which some nodes are more powerful than other nodes. Usually there is a smaller group of more powerful nodes known as cluster heads which would gather data from the less powerful nodes [28].

3D environment specific constraint: In 3D sensor networks, nodes are distributed over a 3D space. The height of the network is not negligible as compared to the length and the width. The nodes may be placed at different heights and depths (in the atmosphere or ocean for example) and the transmission radius of a node may cover or extend over 360 degree.

Thus, 3D sensor networks share a lot of sensor nodes features with 2D sensor networks. The main difference is the dimension of the work space: 3D. This obviously leads to changes in network design and management. However, constraint characteristics like power consumption, computation, and storage capacities or ability to cope

with network failures and changes are much stressed and they influence the routing in these networks. 3D WSNs are as well networks with high resource-constrained and dynamic nature with sensor nodes which have limited memory and inaccurate local information. This has made the design of 3D routing protocols more complicated and challenging.

2.1.2 Network Design Objectives

Designs objectives of 3D sensor network mainly depend on requirements of the underlying sensing application that will be run on it. Thus, some frequent objectives found in wireless sensor network design include the following [26, 27, 29].

- **Many more nodes to cover the 3D space**: 3D WSNs need many more nodes to cover the space. Thus, a network design objective is to build small node size networks. Then, with appropriate coverage technique numbers of deployed nodes can be reduced in order to facilitate node deployment and to minimize the power consumption and cost of sensor nodes. Nodes deployment is dense and 3D algorithm design is based on it.
- **Dynamic network topology**: Any topology change in the network will have an influence on the communications path (routes) between the sensor nodes. Therefore, network design should consider network topology dynamic which is due to node mobility, joining or leaving node, and to node or communication failure.
- **Guaranteed connectivity**: Network connectivity is very important to ensure data routing and maintain an available sensor network. More precisely, any source node that generates data should be connected with the sink otherwise network cannot operate properly.
- **Data redundancy and security**: Moreover, the network design should provide data redundancy in order to increase data accuracy. When multiple sensors sense the data, they ensure better decision making by the sink node and also provide a security level to the network. Therefore, the sensor networks introduce effective security mechanisms to prevent the data information in the network or a sensor node from unauthorized access or malicious attacks.
- **Robustness and adaptability**: Furthermore, the sensor network has to be fault-tolerant. In fact, when there is a failure at node level (deplete all energy, leave the network, etc.) or at communication links level (blockage, interferences, etc.) and when there is a new network node entry (node joining the network), the network should ensure maintenance of all nodes and update all algorithms to fit with the new topology. Hence, sensor network should tolerate the presence of faulty sensors and remain functional in spite of those faulty sensors. This is network design robustness and adaptability properties.

Thus, network protocols designed for sensor networks should be adaptive to such density and topology changes. Radios must be robust in harsh environments and System needs to adapt to various application requirements.

- **Reliability**: Also, network reliability is needed to provide error control and correction mechanisms to ensure reliable data delivery over noisy, error-prone, and time-varying wireless channels.
- **Minimization of energy consumption**: Another desired network property is to extend network lifetime. When sensors nodes exhaust all their energy, they no longer belong to the network. So life expectancy of sensor is related to their energy consumption. And sensor nodes are powered by battery and it is often very difficult or even impossible to charge or recharge their batteries. Therefore, one of the main network objectives is to lower power consumption of sensor nodes so that the lifetime of the sensor nodes, as well as the whole network is prolonged.
- **Node resources constraints**: Moreover, sensor network protocol design should take into account resource constraints of sensor nodes. Thus, running protocols on sensor nodes should not require high computational operations or high capacities storage.

2.2 Routing Challenges and Design Issues in 3D WSNs

Most existing routing algorithms in WSNs assume a two-dimensional topology. Thus, when making the step from 2D to 3D, some issues arise. These issues are discussed in this part.

Deployment and configuration of sensor networks to ensure desired levels of connectivity and sensing coverage is fundamentally more challenging in 3D as compared to 2D [30]. In fact, the coverage of an entire area also known as full or blanket coverage means that every single point within the field of interest is within the sensing range of at least one sensor node. Ideally, it is preferred to deploy the minimum number of sensor nodes within a field in order to achieve blanket coverage. Sensor network entire connectivity is also a challenge in 3D WSNs since the sensing space is larger, nodes are more numerous and nodes can be deployed in a random manner. These factors can create connection interruptions. So, routing protocols must have sufficient connectivity for navigation and forwarding purposes. Then, an issue for routing in 3D WSNs is the network coverage and connectivity. The challenges here are to find the best way to place the nodes in three dimension such that the number of nodes required for surveillance of a 3D space is minimized, while guaranteeing 100 % coverage and to find the minimum ratio of the transmission range and the sensing range of such a placement strategy.

In 3D WSNs the problem in routing scalability and energy efficiency is greatly exacerbated in comparison with its 2D counterpart. This is due to dramatically increased sensor nodes in order to cover a 3D space. Thus, the routing protocol should provide a high scalability using the minimal data and network resources. And, algorithms designed for these 3D Wireless Sensor Networks need to be both memory and energy efficient.

Also, in routing algorithms, computation complexity each node depends on the number of communicating neighbors or the number of nodes with which its sensing

range overlaps. In 3D, both these numbers are twice the corresponding numbers in 2D. Hence routing protocol algorithms are more complex and required more sensor resources (computation, storage, etc.). However, characteristics of sensor nodes require the design of new protocols that take into consideration resources scarcity in sensor nodes like memory and computing power [31].

Another essential criterion that should be taken into consideration while designing a protocol for wireless sensor nodes is power consumption. Since sensor nodes are battery powered, energy becomes a limiting factor. In most cases, changing or recharging the battery might cost more than deploying a new node. Hence, extending the network lifetime is a critical metric in the evaluation of wireless sensor network protocols. That is why traditional routing algorithms like distance vector and link state are not suitable for the use in wireless sensor networks. Moreover, when passing from 2D to 3D WSNs, additional dimension to the destination location leads to more possible routing direction. This has the disadvantage of reducing delivery rate which is a real challenge in routing protocols of 3D WSNs. In fact, most of routing protocols (in 2D) handle delivery using their own assumptions which limit the use of such algorithms to specific environments that satisfy these assumptions. However, these assumptions are not anymore hold in 3D WSNs.

For example, let us consider the local minimum problem. In general graph, data forwarding may be stuck at a node that is a local minimum, i.e., it is closer to the destination than any of its neighbors [33]. In these cases, when a packet is stuck in a 2D WSNs routing protocol, 2D face routing protocol [32, 34] which is the most prominent solution to recover from local-minimum, move the packet out of the local minimum. Then, these protocols provide guaranteed delivery for a planar graph [35]. When using face routing in 2D WSNs, the faces to be traversed are determined by the line from source to destination. However, in 3D graphs, this line does not determine the faces [10] but its boundary becomes a surface, yielding an arbitrarily large number of possible paths to be explored and thus rendering face routing infeasible. In fact in [36], they recently proved that there is no deterministic localized routing algorithm for 3D networks that guarantees the delivery of packets. There is no planar topology concept any more in 3D networks and simple projection from 3D to 2D may break the network connectivity. Thus, 2D face routing algorithms are not directly applicable to 3D. Hence, guaranteed delivery in 3D sensor network is very challenging.

Moreover, in 3D WSNs, routing protocols have much more routing possibilities to the destination of a packet. So, depending on the algorithm the routing or distance stretch factor can increase considerably compared to the shortest path, which leads to a waste of nodes resources. Then, a good routing algorithm should provide efficient stretch factor and storage overhead.

In addition, routing and data dissemination protocols must guard against radio frequency (RF) flooding because too many transmissions in a fixed bandwidth could lead to a serious degradation in RF performance [29]. Moreover, routing algorithms should adapt with sensor network topology continuously changing, self-adapting to the connectivity and propagation conditions and to the traffic and user mobility patterns. Even in harsh propagation conditions, algorithms have to provide reliable and uninterruptible communication. And, since the number sensor nodes in sensor

networks are in the order of tens, hundreds, or thousands, network protocols designed for sensor networks should be scalable to different network sizes. Also, adaptability of routing protocols is needed to handle node failure, joining or moving, which would result in changes in node density and network topology.

Furthermore, routing protocols are based on network information like for example node location. There is a possibility to get inaccurate or even faulty information which may affects the performance and the correctness of the routing protocol. This is due to failures at the nodes, links, or even communication between the sensors. Thus, this may require more robust protocols capable to tolerate faulty or inaccurate data. Similarly, the asymmetry in nodes communication affects the routing protocols since the communication between two nodes is not equal. So 3D routing protocols should be more robust to handle incomplete data due to a lack of information resulting from the asymmetric communication.

And, since sensor networks have limited bandwidth resources, communication protocols designed for sensor networks should efficiently make use of the bandwidth to improve channel utilization. All this should be taking into account in the routing and data dissemination protocols of 3D WSNs. Designing efficient routing protocols in 3D networks is surprisingly difficult, even though similar problems in 2D can be easily handled.

3 Terms and Concepts Definitions of Routing Protocols in 3D WSNs

This part focuses on the basic and general definitions of terms and concepts used in 3D routing and data dissemination.

3.1 Terminology and Models for 3D WSNs (Concepts and Definitions)

Here, we describe Delaunay Triangulation (DT) and Unit Disk Graph (UDG) / Unit Ball Graph (UBG) models that have been widely used as network model of 3D WSNs.

3.1.1 Delaunay Triangulation

A distributed DT of a set S of nodes is specified by $\{< u, Nu > | u \in S\}$, where Nu represents the set of u's neighbor nodes, which is locally determined by u.

A triangulation of a set S of nodes (points) in 2D is a subdivision of the convex hull of nodes in S into non-overlapping triangles such that the vertices of each triangle are nodes in S. A DT in 2D is a triangulation such that the circumcircle of each triangle

does not contain any other node inside [37]. The definition of DT can be generalized to a higher dimensional space using simplexes and circum hyper spheres. In each case, the DT of S is a graph to be denoted by $DT(S)$.

3.1.2 Unit Disk Graph and Unit Ball Graph Diagrams

Assume that the set of n wireless hosts is represented by a point set S in the 3D space. All the network hosts have the same communication range R, which is represented as a sphere volume of radius R. Two nodes are connected by an edge if the Euclidean distance between them is at most R. The resulting graph is called a unit disk graph (UDG). The Gabriel Graph [38] is a sub-graph of the graph G that can be constructed locally as follows: given any two adjacent nodes u and v in G, the undirected edge (u, v) belongs to Gabriel Graph if, and only if, no other node $w \in G$ is located in the sphere of minimum diameter circumscribing (u, v). The Gabriel Graph is planar if G is 2D-UDG. The 3-D UDG is a common geometric graph to represent sensor networks and ad hoc networks. Moreover, since we are dealing with sensor networks, each node has a fixed spherical sensing range Rs.

Then, in 3D space, the disk is replaced by a corresponding three-dimension ball; the obtain graph is the Unit Ball Graph called UBG.

3.2 Taxonomy of 3D WSNs Routing Protocols

We present the different classes of routing protocols mostly used in 3D WSNs.

3.2.1 Position-Based Protocols (Geographic Routing)

Position-based protocols also called location-based protocols are de-fined in the following. Most of the routing protocols for 3D Wireless Sensor Networks require location information for sensor nodes. Generally, location information is needed to estimate the position of a node in the network and at the same time, figure out who are its neighbors and their position. Then, the distance between two particular nodes can be calculated and the energy consumption can be evaluated. Since, there is no addressing scheme for sensor networks like IP-addresses and they are spatially deployed on a region, location information can be utilized in routing data. Diverse kind of routing algorithm for 3D WSNs use this location information to build a path from a source node to a target node. In fact, using nodes position in the routing algorithm brings more scalability and can guarantee packet delivery with less storage. In this way, stringent resource constraints in 3D WSNs on individual node about computation complexity and storage space bounded by a small constant can also be fulfilled. Then, position-based or location-based routing algorithms eliminate some of the limitations of topology-based routing by using additional information.

A location service is used by the sender of a packet to determine the position of the destination and to include it in the packet's destination address. Thus, location-based routing does not require the establishment or maintenance of routes (*Forwarding Strategy*). Several routing techniques are position-aware like greedy routing [39], geometric routing [40], etc. [41, 42]. The next section gives an overview of location-aware routing protocols proposed for 3D WSNs.

In fact, Position-based routing protocols assume that the node knows: (1) the coordinates (*x, y, z*) of its position, which can be obtained using a method like a global positioning system; (2) the location of its neighbors using a periodical exchange of control messages; and (3) the location of the destination, e.g., by using a location service [13]. The position-based routing task is to find a path from the source node to the destination node. It uses the local information at each node to determine how to route the packet. We are interested in the following performance measures for routing algorithms: the delivery rate, which is the percentage of times that the algorithm succeeds in delivering its packet, and the network survivability, which can be measured by the remaining power in the maximum used node during a set of consecutive routing messages.

Geographic routing or Position-based routing (also called geometric routing) is a routing principle that relies on geographic position information. It is mainly proposed for wireless networks and based on the idea that the source sends a message to the geographic location of the destination instead of using the network address. In geographic routing schemes, each network node is assumed to know the coordinates of itself and all adjacent nodes, and each message carries the coordinates of its target. We define a geographic routing algorithm to base its decision solely on the position of the current node, the neighbors, and the destination, and we require the network nodes to be memory-less, i.e., not store any state for messages they see. This not only binds the routing state uniquely to the messages, but also removes an additional storage overhead from the nodes, which could limit the number of messages forwarded by a node if its memory is too small. As a matter of fact, the size of the memory is not the largest challenge. The problem of storing message state is that this data arrives dynamically, and it is hard to predict how much of this data needs to be stored at any given time. Dynamic memory allocation would solve the problem, but introduces an overhead that many devices cannot afford.

Geographic routing has the advantage that it is more scalable due to the lesser need for routing information.

But the routing algorithm may not perform well in case the location of the nodes changes rapidly since the routing table may grow in size and maintaining the routing tables will cause significant overheads.

3.3 Greedy Routing

A greedy algorithm is an algorithm that follows the problem solving heuristic of making the locally optimal choice at each stage with the hope of finding a global

optimum. In many problems, a greedy strategy does not in general produce an optimal solution, but nonetheless a greedy heuristic may yield locally optimal solutions that approximate a global optimal solution in a reasonable time.

Greedy routing is interesting in routing design of 3D WSN for many reasons. With both of its computation complexity and storage space bounded by a small constant, greedy routing is known for its scalability to large networks with stringent resource constraints on individual nodes. Under most greedy routing algorithms, a node makes its routing decision by standard distance calculation based on a small set of local coordinates only. Such salient property is imperatively needed in the emerging 3D sensor network, where the problem in routing scalability is greatly exacerbated in comparison with its 2D counterpart, due to dramatically increased sensor nodes in order to cover a 3D space.

4 Overview of Routing and Data Dissemination Protocols in 3D WSNs

Routing and data dissemination algorithms in three-dimensional Wireless Sensor Networks are mostly location-based. Thus, these routing protocols should be able to have position of nodes in the 3D WSN. A node can learn its position through hardware support such as GPS. Alternatively, the position can be obtained through localization algorithms, of which a variety has been proposed in recent years [43, 44]. Existing geographic routing algorithms can be broadly classified into two categories:

1. Beacon-based.
2. Beacon-less.

This classification was adopted for simplicity

4.1 Beacon-Based Localized Routing Algorithms

Beacon localization approaches have been proposed in the literature as an interesting alternative for centralized/decentralized approaches because of their low cost, their accuracy and their low energy consumption. Beacon-based algorithms require the forwarding node to know the position of all neighbors in its transmission range. Position of the direct neighbors is obtained by observing beacon messages (hello-messages). Each node periodically broadcasts beacon messages which containing its own position and identifier. However, their main drawback in a random deployment remains the lack of a well-defined mobile beacon trajectory that ensures localization for all the nodes with acceptable error estimations.

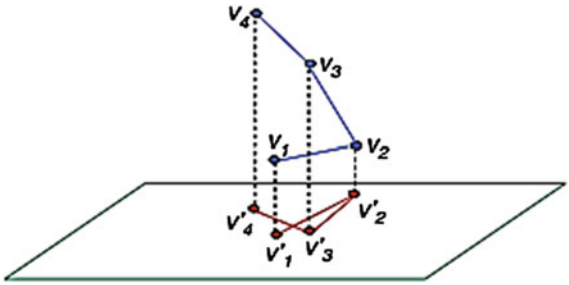


Fig. 1 Projective face routing algorithm. The neighboring nodes are preserved after projection [45]

4.1.1 Projection Heuristic

Projection heuristic: The Position-based routing protocol [45] is a geometric routing protocol primarily proposed for 3D MANETs, but can also be used for 3D WSNs because it favors energy conservation.

The design of this algorithm is motivated by Face routing. Face routing [46] guarantees the delivery on 2D geometric planar graph. The line *st* that connects the source and destination nodes determines the 2D faces to be traversed. But, this line does not determine these faces in a 3D graph and this algorithm cannot be directly applicable on 3D graph. Then, the simple heuristic proposed here is a projection of the 3D network to a 2D plane in order to apply face routing. It is an orthogonal projection on the 2D plane as depicted by Fig. 1. However, Face routing on the projected plane does not ensure a packet to move out of a void in the original 3D network so it cannot guarantee the delivery.

4.1.2 Three-Dimensional Circular Sailing Routing

Three-dimensional circular sailing routing (3D-CSR) [12] algorithm is a load balancing greedy routing for 3D WSNs, based on mapping techniques. 3D-CSR proposes other projection methods to map nodes in 3D space on a sphere. The projection methods guarantee the one-to-one mapping between nodes in 3D networks and virtual nodes on the sphere. They establish two projection methods to map the nodes in 3D space to a sphere (either a 3D sphere or a 4D sphere) and then route the packets on the sphere.

First, considering a node with the following coordinates *m(x, y, z)* in a finite 3D region IR. This node is mapped to the node $m'(x', y', z', \rho)$ where (x', y', z') represents the 3D projection of the projection node m' and ρ is the Euclidean distance from m to the center of the 3D sphere *O (0, 0, 0)* which radius equals r. The virtual coordinates of m' are computed according to the following equations:

$$x' = \frac{r}{\sqrt{x^2 + y^2 + z^2}} x$$

$$y' = \frac{r}{\sqrt{x^2 + y^2 + z^2}} y$$
$$z' = \frac{r}{\sqrt{x^2 + y^2 + z^2}} z \text{ and,}$$
$$\rho = \sqrt{x^2 + y^2 + z^2}$$

Using these virtual coordinates, the geodesic [47] shortest distance d(m′n′) between two projections m′ and n′ on the sphere of the nodes m and n respectively, is calculated according to the following equations:

||mn|| represents the Euclidean distance between the two points *m* and *n* [12].

$$dm'n' = r arcos \frac{om'^2 + on'^2 - m'n'^2}{2om'on'}$$

If m′ and n′ are in different positions on the sphere (Fig. 2a).

$$dm'n' = dmn = mn$$

If m′ and n′ are at the same point on the sphere (Fig. 2b).

For the mapping to a 4D sphere, in the same manner the virtual coordinates are computed and the geodesic shortest distances are deducted. In the case the equations are the following:

$$x' = \frac{4r^2}{x^2 + y^2 + z^2 + 4r^2} x$$
$$y' = \frac{4r^2}{x^2 + y^2 + z^2 + 4r^2} y$$
$$z' = \frac{4r^2}{x^2 + y^2 + z^2 + 4r^2} z,$$
$$w' = \frac{4r^2}{x^2 + y^2 + z^2 + 4r^2} w,$$

And

$$dm'n' = r arcos\left(\frac{x_{m'}x_{n'} + y_{m'}y_{n'} + z_{m'}z_{n'} + w_{m'-r} + (w_{n'-r})}{r^2}\right)$$

Figures 2 and 3 depict the two projection methods respectively.

Then, the 3D-CSR after computing the virtual coordinates of each sensor node and of its neighbors, as well as the geodesic distance for any link, uses this latter calculated distance as the cost on each link (called circular distance) and applies a shortest path algorithm with circular distance as the routing metric to choose the best path minimizing the total circular distance.

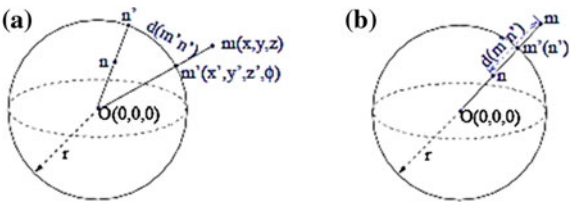


Fig. 2 Projection method I: from a node m(x, y, z) in 3D space to a node m′(x′, y′, z′, ρ) on the 3D sphere, and two case calculations of d(m′n′) [12]

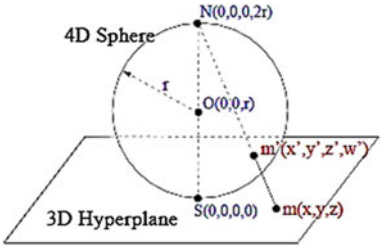


Fig. 3 Projection method II: Stereographic projection: a one-to-one mapping from a node in a 3D hyper-plane to a node on a 4D sphere [12]

Moreover, a localized-based version of the 3D-CSR called 3D-LCSR (Localized Circular Sailing Routing) permits to route packets only on the base of local information at each node. This formulation brings to the algorithm low overhead, easy implementation and good scalability [12]. Even if, 3D-CSR uses another projection scheme specially proposed for load balancing and can reduce hot spots of congestion (wireless congestion) in the network and increase the energy lifetime of the network by spreading the traffic across a virtual sphere, it does not guarantee delivery.

4.1.3 Greedy Distributed Spanning Tree Routing for 3D Sensor Networks

The Greedy distributed spanning tree routing for 3D sensor networks (GDSTR-3D) routing protocol is a geographic routing based on GDSTR [48] protocol for 2D networks, which has been extended and adapted for 3D sensor networks. The routing protocol is based on a convex hull-based tree structure. A hull tree is a spanning tree where each node has an associated convex hull that contains within it the locations of all its descendant nodes in the subtree rooted at the node. Then, the convex hulls of the nodes uniquely define a routing subtree that must contain the destination node, if the packet is deliverable. The routing subtree is defined as the subtree comprising of all the nodes in the network whose hulls contain the coordinates of the destination node. If a packet is not deliverable, the routing subtree will be a null tree. The principle of the GDSTR-3D routing algorithm is defined into two forwarding modes: the greedy mode and the tree forwarding mode.

First, GDSTR-3D attempts to forward packets greedily, i.e., to the neighbor whose coordinates is strictly closer to the destination node in Euclidean distance than the current minimum node. GDSTR-3D has also a feature that is a node has records of

its one-hop neighbor information. Thus each node is aware of two-hop information and if none of its immediate neighbors is strictly closer to the destination, it will attempt to forward the packet instead to the neighbor that has a one-hop neighbor that is closer to the destination than the current minimum.

When a local minimum is reached, GDSTR-3D switches to tree forwarding mode by forwarding the packet along the edges of a spanning tree, guiding the packet to escape from the local minimum.

And, GDSTR-3D switches back to greedy when it finds a neighbor that is strictly closer to the destination than the current minimum. GDSTR-3D is geographic routing protocol for 3D networks that uses 2-hop neighbor information during greedy forwarding to reduce the likelihood of local minima, and aggregates 3D node coordinates using two 2D convex hulls. Thus, it offers deterministic routing and is able to guarantee packet delivery. However, each node must maintain a set of convex hulls, and thus requires a storage space proportional to network size and some nodes (such as the roots of trees) are heavily loaded.

4.1.4 Greedy-Hull-Greedy

The Greedy-Hull-Greedy (GHG) [13] is an efficient geometric routing algorithm for 3D sensor networks. GHG is considered as a 3D analog to Face Routing. The algorithm is based on network hulls constructed with partial unit Delaunay triangulations.

Like most of geometric routing protocol, GHG starts with greedy forwarding, which is simple and close to optimal. In greedy forwarding, each node knows the positions of its neighbors, and the node forwards each message to the neighbor that is the closest to the message's destination. However, greedy forwarding is not always successful: it fails when a message reaches a local minimum node whose neighbors are all further away from the destination than the node itself.

Then, they define a localized 3D partial unit Delaunay triangulation (PUDT) algorithm for capturing the empty 3D network subspaces in order to perform an efficient local-minimum recovery search. For a set of vertices in 3D space, applying Delaunay Triangulation (DT) [49, 50] divides the space into a number of non-intersecting tetrahedra and a single outer subspace. A tetrahedron is determined by a set of four points $T(p_1, p_2, p_3, p_4)$. With DT, tetrahedra are constructed such that there is no point inside the circumsphere $T(p_1, p_2, p_3, p_4)$ of which is the ball(p_1, p_2, p_3, p_4). A ball$(p_1, p_2, p_3, p_4)$is determined by four points not in the same plane. Thus, PUDT algorithm is used to remove intersecting triangles in a way that there is no intersecting edge and triangle, then there is no intersecting tetrahedra. After the PUDT algorithm, each node knows all of its adjacent valid edges and triangles. Then, each node locally groups single edges and triangles into hulls of different subspaces (i.e., identify local hulls) [13] and for a given destination, uses hull-based routing to select the target [13]. Determine the target hull consist to find a representative object (a triangle or a single edge) of the target hull that is the closest object to the (s, t)-segment.

So, analogous to Face Routing, once the message reaches a local minimum, hull-based routing is used to partition the network into subspaces, limiting the recovery to

search a subspace only. This process constrains the local minimum recovery search on the hull of a particular subspace instead of the whole network. GHG algorithm switches to greedy mode when a node that is closer to the destination than the local-minimum is reached.

An execution of GHG like some other geometric algorithms is a repetitive alteration between greedy forwarding and hull-based local-minimum recovery. Here, Delivery is guaranteed since hull routing can always make progress. However, like planarization, the distributed computation of Delaunay triangulations is a hard problem [51] and so GHG is not likely to be usable in practical networks with arbitrary topologies.

4.1.5 Deterministic Greedy Routing Based on a Unit Tetrahedron Cell Mesh Structure (UTC-Greedy Routing)

In Greedy routing algorithms, a node makes its routing decision by standard distance calculation based on a small set of local coordinates only. These greedy routing algorithms called also node-based greedy routing have a computation complexity and a storage space bounded by a small constant, and a well-known scalability to large networks with stringent resource constraints on individual nodes. Such properties make it very attractive for 3D Wireless Sensor Networks. Nevertheless, this kind of algorithms bring challenging issues when there are used for 3D space. The problem of local minimum (void), that is no longer a face but a surface where the possible paths which have to be explored, is very large.

Such local minimums may appear at either boundary or internal nodes. Also, greedy routing in 2D cannot be extended for 3D networks. More, the challenge of greedy routing in 3D networks is further revealed in [36], which proves that there does not exist a deterministic algorithm that can guarantee delivery based on local information only in 3D networks. UTC-greedy routing in [52] investigated decentralized solutions to achieve greedy routing in 3D sensor networks.

This solution is based on a Unit Tetrahedron Cell (UTC) mesh structure. A UTC is a tetrahedron formed by four network nodes, which does not intersect with any other tetrahedrons. They design a simple iterative algorithm to create a mesh UTC from the 3D network.

The objective of the UTC-Greedy Routing is to enable greedy routing from any source to any destination in a given 3D sensor network.

Then, for internal UTCs (local minimum at internal node), a Faced-based greedy routing is used to establish the route from a source to a destination node. The source node computes a line segment between s and d the destination node. This line denoted ϕ passes through a set of UTCs between s and d, and intersects with a sequence of faces where data packets are forwarded and which determine the path of the message.

The previous Faced-based greedy routing cannot support greedy data forwarding at boundaries when there is a local minimum. Then, a distributed algorithm has been proposed to realize volumetric harmonic mapping (VHM) under spherical boundary condition. It is a one-to-one map that yields virtual coordinates for each node in the

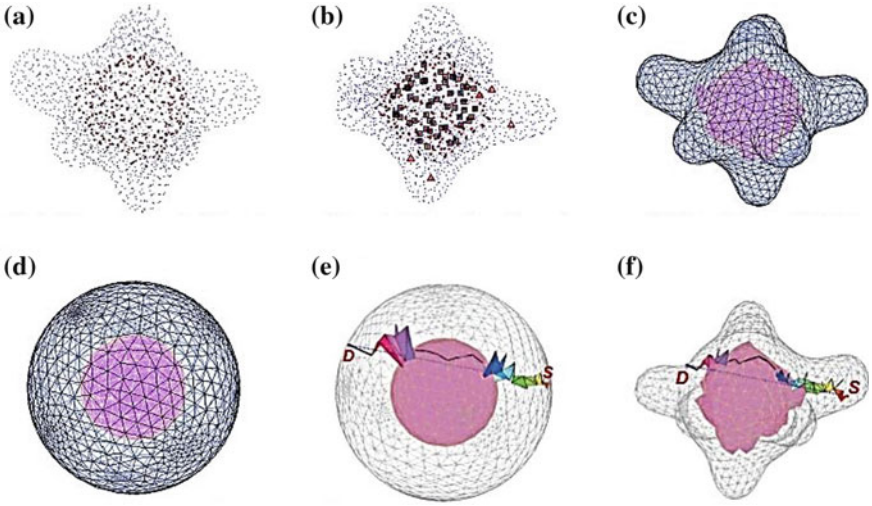


Fig. 4 Illustration of the proposed greedy routing protocol [52] **a**, **b**, **c**, **d**, **e**, **f**

entire 3D wireless sensor network to enable global end-to-end greedy routing. In fact, the entire UTC mesh is homeomorphically (one-to-one) mapped to a solid tetrahedral ball in IR^3 [52]. After the boundary has been mapped to a sphere, node-based greedy routing is always successful in this case.

Thus, during network initialization, the above mapping algorithm is executed on the UTC mesh, so that each node has its own virtual coordinates in a 3D space. Then to route a data packet to its destination, face-based and node-based greedy routing algorithms are employed alternately at internal and boundary UTCs, respectively. The source node first identifies a sequence of faces F that intersects with the line segment between s and d. If the next face is reachable according to local information, the packet is forwarded accordingly by face-based greedy routing. When the packet fails to find the next face toward node d , it must arrive at a boundary, which has been mapped to a sphere. Thus node-based greedy routing is applied to move the packet across the void. Whenever d becomes reachable, face-based greedy routing is applied again. The above process continues until the packet reaches its destination. Figure 4 depicts the greedy routing protocol.

The UTC-Greedy realizes deterministic greedy routing with constant-bounded storage and computation in 3D sensor networks ensuring de livery.

4.1.6 Three-Dimensional Greedy Anti-void Routing

Greedy-based routing algorithms within the three-dimensional (3D) Wireless Sensor Networks face the unreachability problem (i.e., the so-called void problem or local minimum) resulting from the greedy routing algorithms that have not been fully

resolved, especially under the 3D environment. Three-dimensional Greedy anti-void routing (3D-GAR) location-based protocol [53] is proposed to solve the void problem under the unit ball graph (UBG) settings. It employs the 3D rolling-ball UBG boundary traversal (3D-RUT) scheme to guarantee the delivery of packets from the source to the destination node.

The 3D-GAR protocol is a hybrid scheme consisting of both the greedy routing algorithm and the 3D rolling-ball UBG boundary traversal (3D-RUT) scheme. The 3D-RUT algorithm is utilized to determine the boundary node set within the networks under the occurrence of void nodes. As the greedy algorithm fails due to the void nodes, the 3D-RUT scheme can be utilized to escape from the void nodes by traversing the boundary node set and finally restart the greedy for-warding process again. In this way, the packet delivery can therefore be guaranteed.

The 3D-RUT scheme is employed at a node where a void problem is detected means a node that cannot continue to use the greedy forwarding algorithm to transmit packet. 3D-RUT therefore, assign a rolling ball that is a 3D ball hinged at the vertex node with half of transmission range as radius, which can freely rotated and define without any network node inside the ball. The corresponding center point of the ball is called the starting point (SP), S_i. It is noticed that there should always exist an SP for each void node. Based on the rotation of the rolling ball, an SP surface is generated with the accumulation of the corresponding SPs to each node in the network. The resulting SP surface will become a constrained SP surface and is stuck with all the surface-adjacent nodes of node i. These surface-adjacent nodes can be served as the next hopping nodes of node i. Subsequently, i node will inform these surface-adjacent nodes to continue the 3D-RUT scheme by sending control packets that contain the information of their corresponding SPs in order to construct other constrained SP surfaces. Repeatedly, all surface-adjacent nodes continue the 3D-RUT scheme until reaching the destination node. As a result, all the constrained SP surfaces are established and identified; they can be aggregated into a closed surface. This closed surface consists of a set of corresponding nodes, which is established as a boundary node set by adopting the 3D-RUT scheme and which define the route. They theoretically probed that the void problem is solved by the 3D-GAR protocol with guaranteed packet delivery.

4.1.7 Greedy Random Greedy

Authors in [36] proved that there does not exist a deterministic local routing algorithm for 3D networks that guarantees delivery of messages. To address this issue, the randomized geographic routing algorithm called Greedy Random Greedy (GRG) routing [54] for UBG (unit ball graphs) networks has been proposed. This algorithm is based on random walk. We recall that greedy geographic routing falls into the local minimal issue and as it has been proved in [36], there is also no deterministic recovery algorithm that could take off packets out of local minima. Then an excursion to random walks helps to escape such local minima.

On a graph $G = (V, E)$, a Random Walk (RW) can be captured using a Markov chain or a flow in an electrical network obtained from G by replacing every edge by a resistance of 1 ohm(A survey on Random Walk in [55]). In order to ensure the performance of the Random Walk, four techniques are applied. The Region Limited Random Walks technique permits to reduce the exploration of the entire network to the ball of around at the local minimum u node of radius k, with the length of the shortest path connecting and the next node v. Since the next node v in the routing is not known in advance as well as the length k, they consider radius of value 2^i, $for\, i = 1, 2, 3...$ until a node closer to the target is found. Another restriction of the RW takes into account only nodes delimiting the hole which causes the local minimum and which needs to be surrounded. Also, building a sparse graph of the network permits to remove edge that is not critical for the connectivity of the graph. And they apply the power of choice for RW which consists to not return to a previous node (a node that is already in the path of the packet).

The principle of the Greedy-Random-Greedy (GRG) in 3D WSN is to forward greedily the packet until a local minimum is encountered. The randomized recovery algorithm (Random Walk—RW) permits to resolve the problem of local minimum by finding a node more close to the destination node. And then, the greedy process can restart again.

Thus, it has been showed that if k is the length of the optimal path between a given pair of source and destination nodes, then the expected length of the route obtained by any randomized or even deterministic routing algorithm is given by $O(k^3)$. Even if the local structure employed by RW permits to escape from voids when a local minimum is reached, such attempts for randomized recovery of local minimums are non-deterministic and often lead to high overhead or long delay. Nevertheless, Random-Walk is currently one of the truly greedy routing schemes with constant-bounded storage and computation complexity.

4.1.8 Three-Dimensional Geographical Routing

Three-dimensional geographical routing is a position-based routing algorithm [31] that makes use of the position to route packets from sources to destinations with high path quality and reliability. It pro-vides high adaptability to changes in topology and recovery of link failures which increases its reliability. 3DGR assumes as in previous algorithms that links between nodes are Bidirectional and does not assume radio ranges are uniform and that they cover unit disks. Three-dimensional geographical routing (3DGR) routing protocol relies on greedy forwarding and geographical routing. Every node knows its position (either through GPS or after running a localization algorithm) and tries to route packets in the direction toward the destination location. Geographic routing has the advantage that it is more scalable due to the lesser need for routing information. A common problem that faces this kind of algorithms is localization errors in addition to other problems related to the specific approach used. The main problem that faces greedy approaches is the void problem. Void problem also called local minimum arises when there is no node closer to destination than

the sender and thus results in failure of the greedy approach in finding a path to the destination although one might exist.

3DGR uses techniques: geocasting, recent path, and battery awareness measures. The purpose of geocasting is to send a message to nodes in a specific geographic region in other words, to check whether a node belongs to a specific region in the 3D space. This method defines an angle which determines the aperture of the geocasting region. The geocasting region is the intersection of source range sphere and the cone whose head is the source and its head angle α.

Recent path technique permits to store recent path to destination locally and temporally in order to forward a packet to the same destination directly using the recent path without applying any algorithm. Hence, significant overhead and delay can be saved.

Recent studies [56] states that the energy consumed from a battery is not equivalent to the energy dissipated in the device. When discharging, batteries tend to consume more power than needed, and can reimburse the over-consumed power later. The process of the reimbursement is often referred to as battery recovery. This behavior is due to chemical characteristics of batteries. So if the battery is given time to recover, its energy can be used more efficiently.

Based on a discrete time battery model, an optimization to 3DGR protocol is to dynamically schedule routing in sensor networks. This algorithm is aware of the battery status of network nodes and schedules recovery to extend their lifetime. This process corresponds to battery awareness measures.

Then, in 3DGR, an initialization phase permits each node to broadcast Hello packet including their position information. At the end of the initialization phase, every node knows its neighbors and their respective position.

When a source wants to send a packet to some destination, it starts by checking if it has a recent path to that destination. If such a path exists, the packet is forwarded to the next node in the path. Otherwise, it geo-casts a small request packet that includes the coordinates of the destination and setting a timer. Then, each node that has heard the request, checks if it is in the intended region specified by the request packet. If not, it silently drops the packet. Otherwise it checks for a recent path to the requested destination. If a recent path exists, it sends a response for the request indicating that. Otherwise, it checks its neighbors' list and picks the closest one to the requested destination. A flow cart of the sending algorithm can be found in [31]. Even if the 3DGR routing algorithm does not guarantee delivery, it provides a way of shortcutting where a path to the same destination is found at an intermediate node. When such a path exists, the algorithm switches to another mode where there is no need for routing anymore. It rather follows the already existing path and therefore the overhead incurred in the routing process is avoided. Also, with the battery-aware energy efficient schemes, the overall lifetime of the network increases.

4.1.9 Energy-Efficient Restricted 3D Greedy Routing

Energy-efficient restricted 3D greedy routing (ERGrd) [57] is a refined 3D greedy routing protocol to achieve energy-efficiency of its paths with high probability. ERGrd is an energy-efficient localized 3D routing method. This position-based routing protocol is based on a variation of classical greedy routing and an extension of a localized routing method [58] designed for 2D networks. The selection criterion of a node in 3D greedy routing being its distance to the destination, ERGrd needs to ensure energy-efficiency of the overall route and of local links. Therefore, this routing method uses two concepts to refine the choices of forwarding nodes in 3D greedy routing [59].

Energy mileage corresponds to the ratio between the transmission distance and the energy consumption of such transmission. A high value of energy mileage corresponds to a transmission over a long distance that consumes few among of energy. Then, the 3D localized routing greedily selects the neighbor who can maximize the energy mileage as the forwarding node.

The second method is the restricted region which aims to reduce the region where a forwarding node can be selected. Thus, the region is defined inside a 3D cone such that the angle between the forwarding node v, the current node u, and the destination node $t\delta(vut)$, have to be less than a certain angle parameter $\alpha \leq \pi/3$;

These two properties ensure the path efficiency of 3D ERGrd routing protocol. Moreover, the study proves that 3D greedy routing guarantee the delivery of packets under the condition that the underlying topology is Delaunay Translation. So, they investigate the asymptotic Critical Transmission Radius (CTR) for 3D greedy routing to ensure the packet delivery in large-scale random 3D sensor networks. A superior bound of the CTR in the case where nodes are generated by a Poisson point process of density over a convex compact region of unit-volume, has been established for general case and for 3D ERGrd algorithm. They theoretically prove that the CTR for 3D greedy routing is asymptotic almost surely (a.a.s)$3\frac{\sqrt{3\beta_0 \ln n}}{4\pi n}$, where $\beta_0 = 3.2$. So, this theoretical result answers a fundamental question about how large the transmission radius should be set in 3D networks, such that the greedy routing guarantees the delivery of packets between any two nodes.

ERGrd consists of the design of 3D greedy routing protocols which can guarantee delivery of packets and/or energy-efficiency of their paths with high probability in a randomly deployed 3D sensor network.

4.1.10 Geometric STAteless Routing

The resource-constrained and dynamic nature of 3D Wireless Sensor Networks, such as nodes have limited memory and inaccurate local information, has made the design of 3D routing protocols complicated. Geometric routing which only uses the local location information to deliver packets with low communication and storage overheads appear to be more suitable. Geometric STAteless Routing (G-STAR) for

3-D Wireless Sensor Networks is a distributed stateless geometric routing protocol [60]. The main idea of G-STAR is to distributively build a location-based tree and find a path dynamically such that no state information is proactively maintained at each node (stateless).

Compared of other 3D geometric routing protocols like GDSTR [61], GHG [13], G-STAR run in only one mode and guarantees packet delivery even when the location information at some nodes is inaccurate or missing. More, this routing protocol does not assume network models such that unit disk model or unit ball model like GRG [54]. However, three conditions are required on the network to ensure that the protocol functions properly. Thus, network links has to be bidirectional, the network has to be connected—any source and destination are connected, and network topology remains static during the routing process since no routing protocol can guarantee packet delivery if the network topology changes fast during the routing process [36]. These assumptions can be fulfilled easily in real-world applications.

In G-STAR, a node always routes a packet to the neighbor closest to the destination as long as no loop is created. To avoid loops, a list of a subset of nodes which a packet has traversed is recorded by the protocol. Thereby, by examining the partial explored-node list of a packet and the locations of neighbors, a node is able to determine where to forward the packet. When a packet is first generated, the partial explored node list is initialized to be empty. When a node either generates or receives a packet, it first appends itself to the partial explored-node list and checks if it appears on the list more than once. If a node appears on the list twice, the nodes in between two entries are one of the branches the packet has just visited. There is no need to keep both entries in the list, so the earlier duplicated entry is kept and the new entry is removed. In other words, a node does not append itself to the list if it is already in the list. By doing so, a node records where the packet originally came from in the partial explored-node list as the last neighbor ahead of it in the list, which is called parent. At this moment, if the node still has neighbors not in the partial explored node list, it forwards the packet to the one closest to the destination among them. Otherwise, the node will just forward the packet back to its parent.

To improve the behavior of G-STAR, a post optimization technique, namely Path Pruning (PP) [62], is introduced. The routing performance of geometric routing protocols improves dramatically at critical network densities since PP permits a node to listen to the wireless radio channel after it transmits a packet and then it can improve the routing of a packet being aware of new changes. Also, to further minimize the overhead of path pruning, a light-weight path pruning, namely Branch Pruning (BP) that goes with G-STAR has been proposed [60]. In BP, if a node forwards a packet to a neighbor different from its greedy choice, it keeps a next hop entry for the destination of that packet and it will then forward the subsequent packets for that destination directly to this recorded next hop.

G-STAR routing protocol builds a location-based tree on-the-fly and finds a short path when traversing the tree. It is a robust protocol in the sense that it functions well even when the location information is inaccurate or not available for some nodes in

the network. The lightweight BP algorithm and the optimization technique PP help to reduce the path length and to optimize the G-STAR obtained path respective.

4.1.11 Power Adjusted Greedy Algorithm

A crucial problem in sensor networks is finding an efficient and correct route between a source and a destination; however for many networks, a more important problem is providing an energy efficient route because of, for example, the limited battery life of the wireless nodes [63]. In position-based routing algorithms, the nodes use the geographical position of the nodes to make the routing decisions. Power adjusted greedy (PAGs) are localized power-aware 3D position-based and beacon-based routing algorithms that increase the life-time of the network by maximizing the lifetime of the nodes. The idea of the algorithm is to replace the constant transmission power of the node with an adjusted transmission power during two stages—first a low power while discovering the neighboring nodes, and, if needed, a second high transmission power during the routing process.

The network model used in PAG algorithms ensures the same communication range of nodes R and builds a Unit Disk Graph (UDG). This power aware routing protocols are based on the adjustments of the node transmission power at two stages—(1) while discovering the neighboring nodes; and (2) during the routing process.

We can distinguish 3 different versions of PAG routing protocols. We have:

1. *Power Adjusted Greedy algorithm (PAG)*: This algorithm can be summarized as follows:

- All nodes use the low transmission range μ, which equals half of their maximum transmission range, to discover their neighbors. This process is done periodically.
- Greedy routing is started between the source and the destination.
- If the packet reaches a local minimum (packet stuck at a node that does not have a neighbor that makes progress to the destination) at low transmission level, then the current node increases its transmission range by a factor of β and runs neighbor nodes discovery step again. Figure 5 gives an example of this point: when the

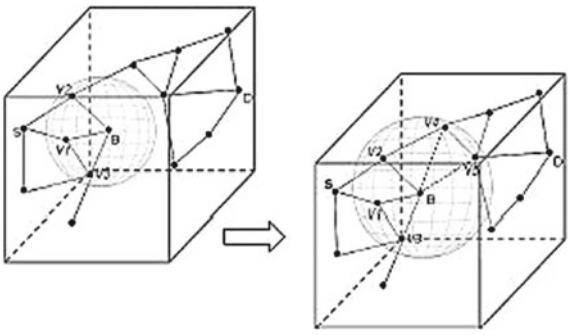


Fig. 5 A node may increase its transmission range in PAG algorithm [63]

message arrives to the node B that does not have any neighbor that make a progress to the destination, B will increase its transmission range by a factor of β to find a new neighbor. Each node can adjust its transmission range just one time while routing a single packet.
- If the node does not discover a new neighbor that makes progress to the destination, then the algorithm fails, otherwise Greedy routing continues

2. *PAG:CFace(3)*: The previous algorithm PAG and its associated fixed power Greedy has a great advantage in terms of power saving. In our simulations it suffers from a low delivery rate if the network is very sparse. The proposed solution is to use CFace(3) (Coordinate Face) [65] routing if the PAG algorithm fails to deliver the message. The combination is called PAG:CFace(3) and can summarized as follows:

- The algorithm starts with PAG routing algorithm.
- If the current node adjusts its transmission range, and after that, it stays in the local minimum situation, then the algorithm changes to CFace(3) algorithm.
- If CFace(3) fails to deliver the message the algorithm fails.

3. *PAG: CFace(1)*:PAG The only difference between this algorithm and PAG: CFace(3) is that instead of trying another projective plane if the first projective plane fails, it returns immediately back to the PAG algorithm. PAG:CFace(1): PAG is summarized in detail as follows:

- The second hybrid algorithm starts with PAG algorithm.
- Once it arrives at a local minimum, and the adjusted transmis-sion range does not help to get a new neighbor to make a pro-gress to the destination, the algorithm switches to CFace(1). CFace(1) is a simplified version of CFace(3), which attempts face routing with the points projected once only onto one of the xy, yz, or xz planes, randomly chosen.
- CFace(1) traverses one projective plane, which is randomly one of the xy, yz, or xz planes starting from the local minimum C as the new source node.
- If the destination is not reached during CFace(1) and looping occurs, the algorithm goes back to PAG.

PAG(s) provide a maximize delivery rate nearly 100 %. Also, with the energy-aware design, there is an improvement of network life time.

4.1.12 3D Sensing Spheres Close the Line Routing Algorithm

3D sensing spheres close the line routing algorithm (3DSSL) protocol focus on two issues in 3D WSNs: coverage and routing. For coverage problem, they introduce a new approach for obtaining a static covered network in the 3-D environment. This technique is referred to as the Chipset model. This would be accomplished by using a small number of sensor nodes in order to save up some energy. For routing issue, a new position-based routing protocol referred to as the 3-D Sensing Spheres close to

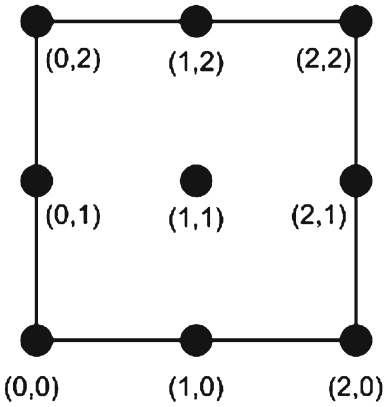


Fig. 6 The top view of the chipset model [66]

the Line routing algorithm (3-D SSL) has been proposed [66]. This protocol achieves 100 % delivery rate on top of the 3-D model. This routing algorithm might reduce the energy consumption of the nodes, therefore, prolonging the lifetime of the network.

At the beginning, the Chipset coverage algorithm constructs a Chipset model like in Fig. 6, consisting of 3-D columns or pins with equal lengths, is only created once for a region, and the pins' densities are neglected. Moreover, the four lines that construct the base (the square holding the pins), do not really exist and they are represented just for illustration purposes. Figure 6 illustrates a Chipset model used to cover a region of an ocean with 2 m $\times$ 2 m $\times$ 2m. Therefore, the height of each pin is 2 m and since the region has a length of 2 m and a width of 2 m along with the assumption that the spacing between the pins, referred to as S, is equal to 1 m, there will be 9 pins constructing the Chipset model.

Then every single point on each of these pins needs to be covered by at least one node's sensing sphere. The algorithm, therefore, starts placing one sensor node at a time at the Chipset model. If a newly placed sensing sphere covers at least one uncovered point on one pin, then the new sensor node becomes active; otherwise it will be discarded. The algorithm keeps placing new sensor nodes until obtaining full coverage for not only for a pin (covering the points on a pin), but for all the Chipset model's pins. This procedure leads to resolve the power saving issue, since they are only placing the necessary sensing spheres at our Chipset model.

Although a full coverage is obtained for all the pins, the model is still partially covered. The reason is that according to the coverage definition mentioned previously, every single point in a chosen region, not only the points on the pins, must be covered by at least one node's sensing sphere. Therefore, there is a possibility that the 3-D spaces between the pins are not covered, since we were only dealing with the pins themselves. We resolve this issue as follows. After covering all the points on all the pins by a set of sensor nodes, the algorithm increases the sensing range by a factor of the spacing between pins S.

After the covering process, 3DSSL protocol can run. This protocol simply works as follows. A current node checks if the sensing spheres of its neighbors intersect the

line segment joining the source and the destination. After having this set of nodes, the current node picks the one which is the closest to the destination. The 3-D SSL algorithm keeps continuing like this until reaching the destination. This algorithm always guarantees the delivery of packets at all the times. This would show that the Chipset model is a robust approach for the 3-D coverage problem.

4.1.13 Geographic Routing MDT

The geographic routing protocol, named MDT [67], is designed for 2D, 3D, and higher dimensions with these properties: (1) guaranteed delivery for any connected graph of nodes and physical links, and (2) low routing stretch from efficient forwarding of packets out of local minima. The guaranteed delivery property holds for node locations specified by accurate, inaccurate, or arbitrary coordinates.

The MDT protocol suite includes a packet forwarding protocol together with protocols for nodes to construct and maintain a distributed MDT graph for routing. Furthermore, MDT protocols are specially designed to handle churn, i.e., dynamic topology changes due to addition and deletion of nodes and links.

Based on Delaunay Triangulation (DT), Multi-hop DT algorithm designs MDT graphs which are the summation of every physical links and every DT edge in the network. MDT is communication efficient. Then, routing is realized with MDT forwarding. The key idea of MDT forwarding at a node, say u, is conceptually simple: For a packet with destination d, if u is not a local minimum, the packet is forwarded to a physical neighbor closest to d; else, the packet is forwarded, via a virtual link, to a multi-hop DT neighbor closest to d.

For a more detailed specification, consider a node u that has received a data message m to forward. Node u stores it with the format: $m =< m.dest, m.source, m.relay, m.data >$ in a local data structure, where $m.dest$ is the destination location,

Table 1 MDT forwarding protocol at node u [66]

Step #	Condition	Action
1.	$u = m.dest$	No need to foward (node u is at destination location)
2.	There exists a node v in Pu and $v = m.dest$	Transmit to v (node v is at destination location)
3.	$m.relay \neq null$ and $m.relay \neq u$	Find tuple t in Fu with $t.dest = m.relay$, transmit to $t.succ$
4.	There exists a node v in $Pu \cup \{u\}$ closest to $m.dest$, $v \neq u$	Transmit to node v (greedy step 1)
5.	There exists a node v in $Nu \cup \{u\}$ closest to $m.dest$, $v \neq u$	Find tuple t in Fu with $t.dest = v$, transmit to $t.succv$(greedy step 2)
6.	Conditions 1–5 are all false	No need to foward (node u is closest to destination location)

m.source is the source node, *m.relay* is the relay node, and *m.data* is the payload of the message. Note that if $m.relay \neq null$, message *m* is traversing a virtual link.

The MDT forwarding protocol at a node, say *u*, is specified by the conditions and actions in Table 1. To forward message *m* to a node closest to location *m.dest*, the conditions in Table 1 are checked sequentially. The first condition found to be true determines the forwarding action. In particular, line 3 is for handling messages traversing a virtual link. Line 4 is greedy forwarding to physical neighbors. Line 5 is greedy forwarding to multi-hop DT neighbors. It has been theoretically proved that MDT forwarding in a correct multi-hop DT provides guaranteed delivery.

4.2 Beaconless Routing Algorithms

In beacon-less routing algorithms, nodes do not require to periodically broadcast the hello (beacon) messages. These algorithms select the next hop forwarder among the previous hop forwarder node's neighbors. But, the previous hop forwarder is unaware of the position and even the existence of its neighbors. Data packets are broadcasted and then using some strategy one of the receiving nodes is selected as a next hop forwarder. Thus, the routing protocol does not require nodes to periodically broadcast hello-messages, and thus avoids drawbacks such as extensive use of scarce battery-power, interferences with regular data transmission, and performance degradation

4.2.1 3D Blind Geographic Routing

BGR is a two-dimensional beacon-less geographic routing algorithm using a broadcast-based contention scheme [68]. 3D Blind Geographic Routing (3D-BGR) extends the routing protocol to work with three-dimensional WSNs.

During BGR functioning, the nodes do not carry any neighborhood or topology information. Packets are forwarded via broadcast. Nodes which receive this broadcast determine if they are located within a special area called forwarding area. A description of the forwarding area is included in the packet. The forwarding area is oriented toward the destination location, and its dimension ensures that all nodes with-in it can mutually communicate with each other (provided the unit disk graph model; however, BGR also performs well with more realistic, irregular radio propagation). Examples for forwarding areas are shown in Fig. 7.

Nodes which receive a broadcast and are located within the forwarding area start a contention timer depending on their distance to the destination. The timer of the node which is closest to the destination expires first; this node declares itself as next hop and forwards the packet again. The other nodes which have still a timer running also receive this packet and cancel their timers.

BGR like Beacon-less algorithms, can easily be extended to operate in 3D space. Thus, the forwarding areas have to be converted into for-warding volumes by constructing the solid of revolution around the forwarder-destination axis.

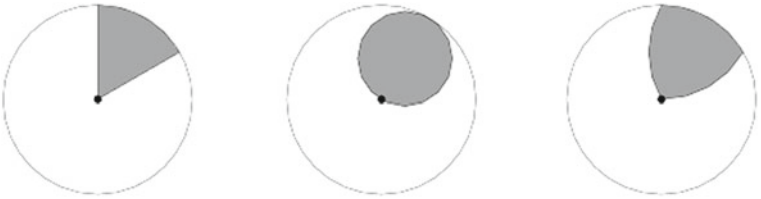


Fig. 7 Forwarding areas: 60° sector, circle, and Reuleaux triangle [68]

Hence, the 2D sector becomes a spherical sector, the circle becomes a sphere, and the Reuleaux triangle becomes the solid of revolution of a Reuleaux triangle. Note that this is different from the Reuleaux tetra-hedron, whose diameter is slightly larger than the radius of the intersecting spheres from which it is constructed.

3D-BGR's forwarding volumes suffer from an analogical problem, since the covered fraction of the transmission volumes is only half as large as in the 2D case. Hence, recovery mode is triggered more often and the number of sent packets increases. The delivery ratio is also slightly lower. Additionally, simulation results revealed that in case of location errors, 3D routing has significantly more problems than its 2D counterpart.

4.2.2 Energy Aware Beaconless Geographic Routing Approach for Three Dimensional WSNs (3D-EABGR)

The proposed energy aware beaconless geographic routing approach for three dimensional WSNs [69] takes into account energy budget of nodes besides the distance, in next hop forwarder selection process.

This increases the lifetime of the network by maximizing the lifetime of the nodes. The assumptions of this routing algorithm are:

1. All nodes are homogeneous in nature. This means, that all the nodes have identical sensing ability, computational ability, the ability to communicate. We also assume that the initial battery powers of the nodes are identical at deployment.
2. All nodes have identical communication range R. Communication is Omni-directional and the communication region of each node can be represented by a sphere of radius R, having the sensor node at its center.
3. Nodes know their own location coordinates through GPS or any other mechanism.
4. Nodes know the location of the final destination.
5. All the nodes are randomly deployed over the surveillance volume.

Then, according to the energy model [69], the forwarding volume is calculated.

In this Algorithm, let e is the energy budget of the node; parameters that are used in the calculation of time interval t are illustrated with the help of Fig. 8. Where d is the distance between the forwarder node and destination, p is the distance between forwarder node and the projection of the candidate node's position on the straight

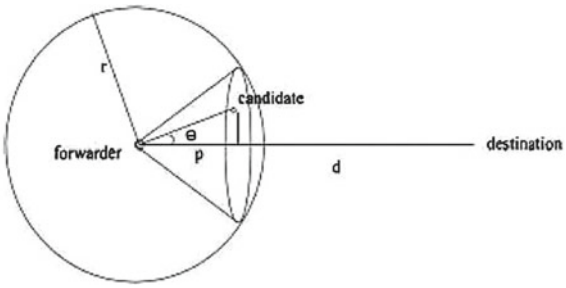


Fig. 8 Parameters for the calculation of the time interval t [68]

line from for-warder to destination and *r* is transmission range. And *MaxDelay* is the maximum time a forwarder is allowed to wait before retransmission.

4.3 Comparison

Although we classified a sample of routing and data dissemination protocols for 3D WSNs, other classification could be made (e.g., on the type or model of networks used, random or deterministic networks [57, 59], and unit ball graph model [53, 54]). The fact is that several efforts have been made to make efficient routing protocols in 3D WSNs. However, all these routing algorithms have strengths and weaknesses.

Regarding our classification, the beacon-based approaches have the following drawbacks:

1. If nodes are mobile then position information of nodes may change over time. This inaccuracy of node's positions increases load on MAC layer. If routing layer is not taking care of this situation packets may be lost.
2. The beacons themselves impose additional load on the network.
3. Beacons can interfere with regular data transmission.
4. Battery power consumption and bandwidth usage by beacons cannot be neglected. Even the nodes that are not taking part in routing process waste their energy and power resource.

Since in beacon-less routing algorithms, nodes do not require to periodically broadcast the hello (beacon) messages, therefore, avoids all the draw-backs of beacon-based algorithms mentioned above. Nevertheless, beacon-less routing algorithms require to be executed under multiple assumptions like:

1. Identical sensor nodes (in sensing and communication abilities); this requirement cannot be easily fulfilled in all three-dimensional sensor networks.
2. Other beacon-less routing simply used broadcast technique which brings large overhead especially in three-dimensional sensor networks where a large number of sensors is needed to cover the entire space. And, the delivery ratio is not guaranteed at 100 %.

The studied routing algorithms are then classified according to some criteria in the following parts.

4.3.1 Network Design

Routing algorithms which use some changing or design new elements to improve or facilitate routing like GDSTR, GHG may introduce an additional overhead compared to the other ones such as ERGrd or G-STAR. In fact, spanning tree is constructed for the entire network (GDSTR) or for a subgraph of the network (GHG) and each node makes use to its children in the tree for routing. All this require high calculations which would demand significant processing resources; it will also introduce high storage over-head in order to maintain the configuration and if the location information is inaccurate, a defective spanning tree will be constructed and routing may fail. These protocols are considered too expensive for 3D WSNs. More such routing that proactively maintains states of information at each node are called statefull unlike the stateless ones like G-STAR or GRG which require less resource capacity.

4.3.2 Network Models

The utilization of simplified models of networks such that Delaunay triangulation (DT), unit disk graph (UDG) and unit ball graph (UBG) allows the routing protocols to search path from source to destination in a uniform network in order to increase to routing efficiency. Network models allow simplifying the network and make it more accessible. However, these models are not always representative of real network especially three-dimensional networks which can be very complex for example in case of configuration designs like undersea 3D WSNs. Most of the routing approaches use models: The Delaunay triangulation (DT) network-based models are used for example in Greedy-Hull-Greedy (GHG), Deterministic greedy routing based on a unit tetrahedron cell mesh structure (UTC-Greedy Routing), Energy-efficient restricted 3D greedy routing (ERGrd) [57], and Geographic routing MDT(Multi-hop DT algorithm), etc. Regarding the algorithms such as Greedy Random Greedy (GRG), Three-dimensional greedy anti-void routing (3D-GAR), or Power Adjusted Greedy algorithm (PAG), the routing protocol is based on a unit ball graph (UBG) settings. Thus, like planarization, the distributed computation of DT or UBG settings is a hard problem [51] and so these algorithms are not likely to be usable in practical networks with arbitrary topologies. More, they require a storage space proportional to network size and some nodes are heavily loaded.

Some routing in return such as Three-dimensional Geographical Routing (3DGR) assumes as in previous algorithms that links between nodes are bidirectional but does not assume that radio ranges are uniform and that they cover unit disks. Similarly for Geometric STAteless Routing (G-STAR), which does not assume network models such that unit disk model or unit ball model like GRG [54] or MDT [67].

Others algorithms like Projection heuristic (PH) [45] and Three-Dimensional Circular Sailing Routing (3D-CSR) [12] use projection to have virtual coordinates or other mapping techniques in order to build a network where the routing protocol can be run. The techniques are very simple leading to low overhead, easy implementation, and good scalability but they cannot guarantee the delivery.

4.3.3 Delivery Rate

The aim of routing is to delivery packets from one source to a destination. But because of 3D topology sensor networks constraints, which often lead to dead ends during the routing of a packet, some routing protocol cannot guarantee the delivery at 100 %. Thus, projection heuristic [45] or load balancing (3D-CSR) [12] are not able to deliver all packets properly since projection methods do not reflect the truly topology of the network. The evaluation of the virtual coordinate is not localized and has to be updated whenever the topology changes. This is the same case for others protocols like 3DGR [31]. Even if the 3DGR routing algorithm does not guarantee delivery, it rather favors other criterion like the overhead which is completely avoided thanks for reuse of the already existing path. Likewise, PAG(s) routing protocols [64] focus on energy-awareness and try to maximize the delivery rate nearly to 100 %. And, Beacon-less routing algorithm like 3D-BGR [68], which forwards packets via broadcasting have a lower delivery ratio since routing decision are taken using the forwarding area extended from 2D to 3D space into a forwarding volume. These forwarding volumes cannot ensure the total coverage of nodes which badly affects the routing success or the delivery rate.

However other random protocol like GRG [54] addresses this issue and may guarantee delivery with certain probability using a random walk which often lead to high overhead or long delay when deterministic methods such as Greedy routing-UTC [52] proposes deterministic routing method that ensures the delivery of all packets. In fact, most of the greedy-based routing algorithms face a delivery problem—unreachability problem also called void problem or local minimum—due to the 3D environment of sensor networks (see Sect. 2.2). Thus several greedy-based routing main objectives, is to handle the void problem in order to ensure a guaranteed delivery. They use network models or routing configuration and assumption which sometimes require high storage and computation complexity. Then some algorithms have been design to offer deterministic routing and guarantee packet delivery. It is the case of GDSTR-3D [61], or GHG [13] algorithm that uses hull routing to progress in the routing protocol. Similarly, the UTC-Greedy routing realizes deterministic greedy routing with constant-bounded storage and computation in 3D sensor networks ensuring delivery. And 3D-GAR location-based protocol [53] under the UBG settings scheme guarantees the delivery of packets from the source node to the destination node. Also, under the condition that the underlying topology is Delaunay Translation (DT), the 3D greedy routing, ERGrd [57] guarantees the delivery of packets with high probability in a randomly deployed sensor network. In addition, routing objective is emphasized in routing algorithm such that 3DSSL which uses the Chipset model and a position-

based line routing protocol to reach 100 % delivery rate for 3D WSNs. Compared of other 3D geometric routing protocols like GDSTR [61], GHG [13], G-STAR [60] guarantees packet delivery even when the location information at some nodes is inaccurate or missing. It is a robust protocol in the sense that it functions well even when the location information is inaccurate or not available for some nodes in the network. As to the geographic routing protocol Multi-hop DT (MDT) [67], it ensures a guaranteed delivery for any connected graph and low routing stretch even if node locations are specified by accurate, inaccurate, or arbitrary coordinates and is designed to efficiently handle dynamic topology changes.

4.3.4 Power Consumption

Another property that should hold 3D WSNs is energy-efficiency. This challenge was pointed explicitly in some algorithm like ERGrd [57] while other ones do not take it as a priority where the goal was more oriented towards guaranteeing reception of packets or ensure load balancing. Therefore, routing and data dissemination protocols designed for WSNs should be as energy efficient as possible to prolong the lifetime of individual sensors, and hence the network lifetime.

Energy-efficient restricted 3D greedy routing (ERGrd) [57] achieves energy-efficiency of its paths through the selection criterion of node based on energy mileage and the Restricted region concepts which allow choosing the best next hop according to the minimum power consumption. It can guarantee delivery of packets and energy-efficiency of their paths with high probability in a randomly deployed 3D sensor network. Similarly, the battery awareness measure of 3DGR [31] ensures the minimum power consumption and increase the overall lifetime of the network. Other dedicated routing algorithms and data dissemination techniques to energy-awareness are the power adjusted greedy algorithms (PAGs) [64] are localized power-aware 3D position-based and beacon-based routing algorithms that increase the lifetime of the network by maximizing the lifetime of the nodes. Furthermore, Energy aware beaconless geographic routing approach for three dimensional WSNs (3D-EABGR) [69] takes into account energy budget of nodes besides the distance, in next hop forwarder selection process. This increases the lifetime of the network by maximizing the lifetime of the nodes. Thus, routing and data dissemination algorithms for 3D WSNs have to be energy-aware design routing protocols in order to deal with severe power consumption constraints.

4.3.5 Modes of Protocols

Another feature that emerges from this study is the different stages or modes used in the routing algorithms. Most of the routing and data dissemination algorithms can use several modes according to routing criteria. Generally, the position-based routing protocols for 3D WSNs use a simple protocol like a greedy forwarding to transmit packets and when the routing algorithms face a void problem, they switch

to another mode which permits to handle the problem and, after the void have been solved; they switch back to the first routing algorithm. Theses algorithms employ alternately routing techniques to route the packets towards their destination.

Therefore, GDSTR-3D [61] routing algorithm is defines into two forwarding modes: the greedy mode and the tree forwarding mode (when there is a local minimum) similarly to GHG [13] which execute a Greedy-Hull-Greedy forwarding protocol by a repetitive alteration between greedy forwarding and hull-based local-minimum recovery. Likewise the GRG [54] follows a Greedy-Random-Greedy routing process. The 3D-GAR [53] is a hybrid scheme consisting of both the greedy routing algorithm and the 3D rolling-ball UBG boundary traversal (3D-RUT) scheme. And UTC-Greedy Routing algorithm [52] route a data packet to its destination using a face-based and a node-based greedy routing algorithms that are employed alternately at internal and boundary UTCs, respectively.

Compared to the previous 3D geometric routing protocols like GDSTR [34] and GHG [13], G-STAR [60] runs in only one mode. And more, it does not assume network models such that unit disk model or unit ball model like GRG [54].

Furthermore, some algorithms do not have mode but their routing and data dissemination process are based on several techniques that should be perform in order to ensure a correct routing. It is the case for 3DGR algorithm [31] which uses geocasting, recent path and battery awareness measures techniques to transmit packets; the Energy-efficient restricted 3D greedy routing [57]: this routing method uses two concepts to refine the choices of forwarding nodes in 3D greedy routing: Energy mileage and the restricted region.

In addition, to improve their behavior, some algorithms have post optimization techniques. G-STAR can improve the routing of a packet being aware of new changes via the Path Pruning (PP) [62], and further minimize the overhead of path pruning, through a light-weight path pruning, namely Branch Pruning (BP) [60].

Therefore, some protocols result in a very practical and good performance in terms of energy-efficiency, network overhead, delay, and data delivery while others routing can offer poor performance in reality. In Table 2, there is a summary according the verified criteria of all these protocols we have presented above.

5 Conclusion and Future Research

Routing and data dissemination protocols for three-dimensional Wireless Sensor Networks received recent attention from the community. Regarding the challenges that these routing protocols should offer in terms of performance, resource conservation and energy efficiency, several routing algorithms and techniques have been investigated. Most of these routing algorithms are location-based and therefore they are subject to related problems like local minimum, position inaccuracy. Authors tried to resolve these issues using diverse methods and they produce acceptable solutions. However, improvements have to be done in energy-efficiency field since energy resource of sensor is scarce. Furthermore, a lot of research has been

Table 2 Comparison summary of 3D WSNs routing protocols

Classification criteria	Protocols
Beacon-based location-awareness	Projection heuristic [45], 3D-CSR [12], GDSTR-3D [61], GHG [13], UTC-Greedy Routing [52], 3D-GAR [53], GRG [54], 3DGR [31], ERGrd [57], G-STAR [60], PAG [63], 3DSSL [65], Geographic routing MDT [66]
Beacon-less	3D-BGR [67], 3D-EABGR [68]
Random routing	GRG [54], 3D-BGR [67]
Deterministic routing	UTC-Greedy routing [52], GDSTR-3D [61], GHG [13], 3D-GAR [53]
Guarantee delivery	GDSTR-3D [61], GHG [13], UTC-Greedy Routing [52], 3D-GAR [53], ERGrd [57], G-STAR [60], 3DSSL [65], Geographic routing MDT [66]
Energy-awareness	ERGrd [57], 3DGR [31], PAG [63], 3DSSL [65], 3D-EABGR [68]
Low overload/ Over-head	Projection heuristic [45], 3D-CSR [12], 3DGR [31], G-STAR [60]
High overhead/ Long delay	GDSTR-3D [61], GHG [13], GRG [54]
Constant-bounded storage and computation complexity	UTC-Greedy Routing [52], GRG [54]
DT-based	GHG [13], UTC-Greedy Routing [52], ERGrd [57], Geographic routing MDT [66] UBG-based GRG [54], 3D-GAR [53], PAG [63]
Virtual coordinates and mapping techniques	Projection heuristic [45], 3D-CSR [12], UTC-Greedy Routing [52]
Two forwarding modes	GDSTR-3D [61], GHG [13], UTC-Greedy Routing [52], 3D-GAR [53]

done in three-dimensional Routing in Underwater Acoustic Sensor Networks [70–72, 15] because of the large variety of oceanographic applications. Even if these routing algorithms are primarily designed for underwater communications, they can be an inspiration in designing routing protocols for terrestrial or aerial 3D WSNs.

References

1. C.F. Huang, Y.C. Tseng, L.C. Lo, The coverage problem in three-dimensional wireless sensor networks, in *Proceedings of IEEE Globecom 2004*, vol. 5 (Dallas, 2004), pp. 3182–3186
2. M. Watfa, S. Commuri, Optimal 3-dimensional sensor deployment strategy, in *Proceedings of 3rd IEEE Consumer Communications and Networking Conference (CCNC 2006)*, vol. 2 (Las Vegas, 2006), pp. 892–896
3. M. Watfa, S. Commuri, The 3-dimensional wireless sensor network coverage problem, in *Proceedings of the 2006 IEEE International Conference on Networking, Sensing and Control (ICNSC'06)* (Ft Lauderdale, 2006), pp. 856–861

4. S.M.N. Alam, Z.J. Haas, Coverage and connectivity in three-dimensional networks, in *Proceedings of the 12th ACM Annual International Conference on Mobile Computing and Networking (MobiCom'06)* (New York, 2006), pp. 346–357
5. V. Ravelomanana, Extremal properties of three-dimensional sensor networks with applications. IEEE Trans Mobile Comput. **3**(3), 246–257 (2004). doi:10.1109/TMC.2004.23
6. F. Li, Z. Chen, Y. Wang, Localized geometric topologies with bounded node degree for three-dimensional wireless sensor networks. EURASIP J. Wireless Commun. Netw. **157**, 2012 http://jwcn.eurasipjournals.com/content/2012/1/157
7. S.M. Nazrul Alam, Z.J. Haas, Topology Control and Network Life-time in Three-Dimensional Wireless Sensor Networks. http://wnl.ece.cornell.edu
8. D. Pompili, T. Melodia, Three-dimensional routing in underwater acoustic sensor networks, in *Proceedings of the 2nd ACM International Workshop on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks (PE-WASUN)* (Montreal, Canada, 2005), pp. 214–221
9. P. Xie, J.H. Cui, L. Lao, VBF: Vector-based forwarding protocol for underwater sensor networks, in *Proceedings of the 5th international IFIP-TC6 conference on Networking Technologies (Networking'06)* (Coimbra, Portugal, 2006), pp. 1216–1221
10. G. Kao, T. Fevens, J. Opatrny, Position-based routing on 3-D geometric graphs in mobile ad hoc networks, in *Proceedings of the 17th Canadian Conference on Computational Geometry, CCCG 2005* (Ontario, 2005), pp. 88–91
11. A. Abdallah, T. Fevens, J. Opatrny, Power-aware 3D position-based routing algorithm for ad hoc networks, in *Proceedings of 2007 IEEE International Conference on Communications (ICC)* (Glasgow, Scotland, 2007), pp. 3130–3135
12. F. Li, S. Chen, Y. Wang, J. Chen, Load balancing routing in three dimensional wireless networks, in *Proceedings of 2008 IEEE International Conference on Communications (ICC)*, (Beijing, 2008), pp. 3073–3077
13. C. Liu, J. Wu, Efficient geometric routing in three dimensional ad hoc networks, in *Proceedings of 28th Annual IEEE Conference on Computer Communications (INFOCOM)* (Mini-conference, Rio, 2009), pp. 2751–2755
14. Y. Wang, C.W. Yi, M. Huang, F. Li, Three dimensional greedy routing in large scale random wireless sensor networks. Ad Hoc Netw. (2011, to appear)
15. J.N. Al-Karaki, A.E. Kamal, Routing techniques in wireless sensor networks: A survey IEEE Wirel. Commun. **11**(6), 6–28 (2004)
16. B. Karp, H. Kung, GPSR: Greedy perimeter stateless routing for wireless networks, in *Proceedings of the 6th ACM International Conference on Mobile Computing and Networking (ACM MobiCom)*, Boston, MA, pp. 243–254, Aug 2000
17. F. Kuhn, R. Wattenhofer, A. Zollinger,Worst-case optimal and average-case efficient geometric ad-hoc routing, in *Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking and Computing (ACM MobiHoc)*, pp. 267–278, Annapolis, Maryland, USA, 1–3 June 2003
18. J. Opatrny, A. Abdallah, T. Fevens, Randomized 3D position-based routing algorithms for ad-hoc networks, in *Proceedings of Third Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (MOBIQUITOUS)*, pp. 1–8, San Jose, California, 17–21 July 2006
19. T. He, J.A. Stankovic, C. Lu, T. Abdelzaher, SPEED: A stateless protocol for real-time communication in sensor networks, in *Proceedings of the 23rd International Conference on Distributed Computing Systems (ICDCS)*, pp. 46–55, Providence, RI, USA, 19–22 May 2003
20. K. Akkaya, M. Younis, A survey on routing protocols for wireless sensor networks. Ad Hoc Networks. **3**(3), 325–349 (2005)
21. I.F. Akyildiz, D. Pompili, T. Melodia, Underwater acoustic sensor net-works: Research challenges. Ad Hoc Netw. **3**(3), 257–279 (2005)
22. X. Hong, M. Gerla, R. Bagrodia, T. Kwon, P. Estabrook, G. Pei, The Mars sensor network: efficient, energy aware communications. in *Proceedings of IEEE Military Communications Conference, MILCOM 2001* (2001)

23. J. Zheng, A. Jamalipour, *Wireless Sensor Networks: A Networking Perspective*. (Wiley, and IEEEE, New York, 2009)
24. N. Attarzadeh, M. Mehrani, A new three dimensional clustering method for wireless sensor networks. Global J. Comput. Sci. Technol., **11**(6) Version 1.0 April 2011
25. http://en.wikipedia.org/wiki/Wireless_sensor_network#Characteristics
26. S. Kumar Singh, M.P. Singh, D.K. Singh, Routing protocols in wireless sensor networks-A survey. Int. J. Comput. Sci. Eng. Survey, **1**(2) (2010)
27. F.M. Al-Turjman, Grid-based deployment for wireless sensor networks in outdoor environment monitoring applications. A thesis submitted to the School of Computing, Queen's University, Kingston, Ontario, Canada, April 2011
28. R. Mulligan, H.M. Ammari, Coverage in wireless sensor networks: A survey. Netw. Protocols Algorithms **2**(2) (2010) (ISSN 1943–3581)
29. F. Morese, Deploying a 3D locator system based on a wireless sensor network architecture. SensNavTM 3D Locator System Alcyone Systems, LLC http://www.alcyonesystems.com
30. S. Poduri, S. Pattem, B. Krishnamachari, G. Sukhatme, Sensor network configuration and the curse of dimensionality, in *Proceedings Third Workshop on Embedded Networked Sensors* (EmNets, Cambridge, 2006)
31. M. Watfa, A. Al Tahan, 3D geographical routing in wireless sensor networks, in *International Conference on, Wireless Networks (ICWN'08)*, pp. 1–7
32. P. Bose, P. Morin, I. Stojmenovic, J. Urrutia, routing with guaranteed delivery in ad hoc wireless networks, in *Proceedings of the International Work-shop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIALM)* (1999)
33. P. Bose, P. Morin, I. Stojmenovic, J. Urrutia, Routing with guaranteed delivery in ad hoc wireless networks. ACM/Kluwer Wireless Netw. **7**(6), 609–616 (2001)
34. B. Karp, H. Kung, Greedy perimeter stateless routing for wireless networks, in *Proceedings of ACM Mobicom* (2000)
35. H. Frey, I. Stojmenovic, On delivery guarantees of face and combined greedy-face routing in ad hoc and sensor networks, in *Proceedings of MobiCom*, pp. 390–401 (2006)
36. S. Durocher, D. Kirkpatrick, L. Narayanan, On routing with guaranteed delivery in three-dimensional ad hoc wireless networks, in *Proceedings of International Conference on Distributed Computing and Networking*, pp. 546–557 (2008)
37. S. Fortune, Voronoi diagrams and Delaunay triangulations, in *Handbook of Discrete and Computational Geometry* ed. by J.E. Good-man, J. O'Rourke, 2nd edn. (CRC Press, New York, 2004)
38. http://en.wikipedia.org/wiki/Gabriel_graph
39. R. Flury, S.V. Pemmaraju, R. Wattenhofer, Greedy Routing with Bounded Stretch. in *Proceedings of The 28th IEEE International Conference on Computer Communications (IEEE INFOCOM)*, pp. 1737–1745 , Rio de Janeiro, Brazil, 19–25 April 2009
40. M. Wattenhofer, R. Wattenhofer, P. Widmayer, Geometric Routing without Geometry, 12th Colloquium on Structural Information and Communication Complexity (SIROCCO), Le Mont Saint-Michel, France, May 2005
41. J. Gao, Location Lased Routing in Sensor Networks II. http://www.cs.sunysb.edu/jgao/CSE590-fall06/Slides/lecture5.pdf
42. M. Mauve, J. Widmer, H. Hartenstein, A Survey on Position-Based Routing in Mobile Ad Hoc Networks. IEEE Netw. Mag. **15**(6), 30–39, 2001
43. W. Ammar, A. ElDawy, M. Youssef, Secure Localization in Wireless Sensor Networks: A Survey. arXiv:1004.3164v1 [cs.CR] April 2010
44. A. Srinivasan, J. Wu, A Survey on Secure Localization in Wireless Sensor Networks, Encyclopedia of Wireless and Mobile Communications, ed. by B. Furht (CRC Press, Taylor and Francis Group, 2007)
45. T.F.G. Kao, J. Opatmy, Position-based routing on 3D geometric graphs in mobile ad hoc networks, in *Proceedings of the 17th Canadian Conference on, Computational Geometry*, pp. 88–91 (2005)

46. P. Bose, P. Morin, I. Stojmenovic, J. Urrutia, Routing with guaranteed delivery in ad hoc wireless networks. Wireless Netw. **7**(6), 609–616 (2001)
47. http://en.wikipedia.org/wiki/Geodesic
48. B. Leong, B. Liskov, R. Morris, Geographic routing without planarization, in *Proceedings of NSDI 2006*, May (2006)
49. J. Gao, L.J. Guibas, J. Hershburger, L. Zhang, A. Zhu, Geometric spanner for routing in mobile networks, in *Proceedings of ACM MobiHoc* (2001)
50. X.-Y. Li, G. Calinescu, P.-J. Wan, Y. Wang, Localized delaunay triangulation with application in wireless ad hoc networks, in *Proceedings of IEEE INFOCOM* (2003)
51. S.S. Lam, C. Qian, Geographic routing with low stretch in d-dimensional spaces. The University of Texas at Austin, Jan., Technical report, 2010
52. S. Xia, X. Yin, H. Wu, M. Jin, X.D. Gu, Deterministic greedy routing with guaranteed delivery in 3D wireless sensor networks, in *Proceeding MobiHoc'11 Proceedings of the Twelfth ACM International Symposium on Mobile Ad Hoc Networking and Computinf*, 2011 Article No.1
53. W.-J. Liu, K.-T. Feng, Three-dimensional greedy anti void routing for wireless sensor networks. IEEE Trans. Wireless Commun. **8**(12), 5796–5800 (2009)
54. R. Flury, R. Wattenhofer, Randomized 3D geographic routing, in *Proceedings of the IEEE INFOCOMM 2008* (2008)
55. L. Lov'asz, Random walks on graphs: A survey. Combinatorics **2**, 353–398 (1996)
56. R. Rao, S. Vrudhula, D. Rakhmatov, Battery modeling for energy aware system design, in *IEEE Computer Society*, pp. 77–87, (2003)
57. Y. Wang, C.-W. Yi, M. Huang, F. Li, Three-dimensional greedy routing in large-scale random wireless sensor networks. Ad Hoc Networks (2010). doi:10.1016/j.adhoc.2010.10.003
58. Y. Wang, W.-Z. Song, W. Wang, X.-Y. Li, T. Dahlberg, LEARN: Localized energy aware restricted neighborhood routing for ad hoc networks, in *Proceedings of IEEE SECON*, p. 46 (2006)
59. M. Huang, F. Li, Y. Wang, Energy-efficient restricted greedy routing for three dimensional random wireless networks, in *Proceedings of the 5th International Conference on Wireless Algorithms, Systems, and Applications (WASA)*, pp. 95–104, Beijing, China, 15–17 Aug 2010
60. M.-T. Sun, K. Sakai, B.R. Hamilton, K. Wei-Shinn, X. Ma, G-STAR: Geometric STAteless routing for 3-D wireless sensor networks. Ad Hoc Netw. **9**, 341–354 (2011)
61. J. Zhou, Y. Chen, B. Leong, P. Sundar, Practical 3D geographic routing for wireless sensor networks, in *Proceedings of SenSys*, pp. 337–350 (2010)
62. X. Ma, M.-T. Sun, X. Liu, G. Zhao, An efficient path pruning algorithm for geographical routing in wireless networks. IEEE Trans. Vehicular Technol. **57**(4), 2474–2488 (2008)
63. A.E. Abdallah, T. Fevens, 1. Opatrny, Power-aware 3d Position based Routing Algorithms for Ad hoc Networks, in EEE International Conference on Communications, 2007. ICC apos;07, Vol. 24–28 June 2007, pp. 3130–3135, June 2007.
64. A.E. Abdallah, T. Fevens, J. Opatrny, Randomized 3-D position-based routing algorithm for ad-hoc networks, in *Proceedings 3rd Annual International Conference on Mobile and Ubiquitous Systems: Networks and Services (MOBIQUITOUS)*, San Jose (2006)
65. T.E. Salti, N. Nasser, Routing in three dimensional wireless sensor networks, in *The IEEE "GLOBECOM" 2008 proceedings*
66. S.S. Lam, C. Qian, Geographic routing, in d-dimensional spaces with guaranteed delivery and low stretch, in *SIGMETRICS'11*, June 7–11, San Jose (2011)
67. M. Witt, V. Turau, *Geographic Routing in 3D*. Hamburg University ofTechnology, Technical report, 2007
68. M. Jain, M.K. Mishra, M.M. Gore, Energy aware beaconless geographical routing in three dimensional wireless sensor networks, in *IEEE, ICAC* (2009)
69. M. Ayaz et al., A survey on routing techniques in underwater wireless sen-sor networks. J. Network. Comput. Appl. (2011). doi:10.1016/j.jnca.2011.06.009
70. J. Heidemann, W. Ye, J. Wills, A. Syed, Y. Li, Research challenges and applications for underwater sensor networking, in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC2006)*, Las Vegas, April, pp. 3–6 (2006)

71. D. Pompili and T. Melodia, Three-Dimensional Rout ing in Underwater Acoustic Sensor Networks in Proc of ACM Int l. Workshop on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks (PE-WASUN), Montreal, Canada, Oct 2005
72. J. Partan, J. Kurose, B.N. Levine. A survey of practical issues in underwater networks, in *Proceedings of ACM International Workshop on Underwater Networks (WUWNet)*, pp. 17–24 (2006)

Part VI
Underground and Underwater Sensor Networks

Chapter 12
The Future of Wireless Underground Sensing Networks Considering Physical Layer Aspects

Agnelo Rocha da Silva, Mahta Moghaddam and Mingyan Liu

Abstract The design of a WSN for the underground environment is typically characterized by an excess of either pessimism or optimism. For many years, underground communication has been considered infeasible. Rather, we should neither abandon the hope of designing a functioning underground WSN, nor expect things to automatically work in an underground setting by simply importing technologies from existing WSNs, the majority of which are developed for aboveground environment. Besides energy challenges (more critical compared to typical WSNs), the design of an underground WSN is governed by the characteristics of the underground communication channel. Compared to over-the-air (OTA) radio frequency communication, signal attenuation in soil can be 20–300 times worse. For instance, a typical communication range of 300 m for a radio transceiver can decrease to less than 1 m in soil. Moreover, while OTA transceivers and underwater communication have been available for many years, the same cannot be said for underground communication. The mining industry has been looking for a long-range, low-power, wireless communication solution for rescue missions in the event of trapped miners due to a collapse, and has so far not been very successful. These facts highlight the challenges in realizing wireless underground communication. Recent innovations based on relatively short-range communication and high density of nodes can potentially lead to the proliferation of wireless underground sensor networks (WUSNs) in the near future. In this chapter, we present in detail the traditional challenges faced by WUSN researchers, the perceived limitations, and recent technological advances that

A. R. da Silva (✉) · M. Moghaddam
Electrical Engineering—Electrophysics, University of Southern California, Los Angeles, CA, USA
e-mail: agnelosi@usc.edu

M. Moghaddam
e-mail: mahta@usc.edu

M. Liu
EECS Department, University of Michigan, Ann Arbor, MI, USA
e-mail: mingyan@eecs.umich.edu

H. M. Ammari (ed.), *The Art of Wireless Sensor Networks*,
Signals and Communication Technology, DOI: 10.1007/978-3-642-40066-7_12,

are beginning to change the outlook. Through this discussion, we show that a generic solution for WUSNs cannot be expected. Instead, the design must be tailored to the application. For instance, the features and techniques to be exploited in designing a WUSN to detect oil pipeline leakage are distinctly different from that of a WUSN for agricultural draught or landslide monitoring.

Wireless Underground Sensor Networks (WUSNs) is a special type of WSN where some (typically, the majority) of the *sensor nodes* are deployed below the ground. Two communication technologies for underground settings have been proposed: radio electromagnetic waves radiation (EM) [1] and magnetic induction (MI) [2]. Although specific WUSN applications can take advantage of one or the other technology, we believe that the future of most WUSNs lies in the strategic integration of both because the drawback of one technology can be compensated by the characteristics of the other technology.

Existing communication models for WUSNs are presented by highlighting the main constraints and the communication performance in typical scenarios. Theoretical results as well as outcomes of real-world implementations are also highlighted in this chapter. Specifically, we start this chapter by introducing radio wave propagation through soil. In doing so, we attempt to better understand the challenges involved in applying traditional WSN hardware and software solutions directly to WUSNs. The pros and cons of the recent EM-based solutions for WUSNs are discussed. Moreover, issues arising from buried antennae and the impact of a phenomenon called lateral waves (LWs) are discussed. We then present WUSN designs based on MI and a comparison between EM and MI solutions. The importance of hybrid solutions based on both EM and MI is highlighted.

This chapter has some similarities with [3]. To illustrate both similarities and differences between the works, the topics of each work are presented in Fig. 1. The topics with the same color are the ones with some similarities. The remaining topics are discussions that are not considered at all or only briefly mentioned in one or other work. In particular, our current work extends [3] by highlighting the potential

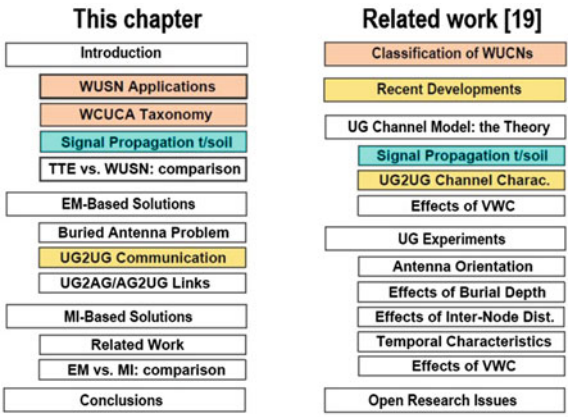


Fig. 1 Comparison between this chapter and Ref. [3]

technological advantages of mixing EM- and MI-based systems. In this context, terms such as LW and UG2AG/AG2UG links are introduced. Also, practical aspects related to the design and deployment of WUSNs are discussed.

1 Introduction

Wireless Underground Sensor Networks (WUSNs) is a special type of WSNs where some of the nodes are deployed below ground, either in soil or in a similar confined environment. For instance, sensors deployed inside walls or in the basement of a building may be considered WUSNs. A variety of novel applications are enabled by the use of WUSNs, initially categorized [1] as follows: environmental monitoring, infrastructure monitoring, location determination, and security monitoring.

The main challenge in the design and operation of WUSNs is the realization of wireless communication given the high EM signal attenuation going through the medium (e.g., soil). For instance, if we consider typical values of soil texture and soil moisture, the communication range between 2 WSN commodity sensor nodes buried at 40 cm depth is smaller than 1 m in the 400–2400 MHz frequency range (10 dBm transmit power level). This estimate is based on the same radio technology used for over-the-air (OTA) communication. Technological enhancement in radio transceiver, antenna design, network protocols, and placement strategies can potentially mitigate this problem [4–11]. However, large coverage using EM-based solution in underground settings remains a challenge. Moreover, various aspects of the soil environment including its makeup, density, and the dynamics of soil moisture, can contribute to significant changes in the conditions of the underground communication channel.

Within this context, MI-based solutions are being proposed as a potential answer especially to large and sparse WUSNs [2]. Given similar volume, cost, and energy parameters, the communication range between two underground MI-based nodes can easily double that of EM-based nodes. More recent development in this area indicates the potential of increasing the range by almost 2 orders of magnitude. Moreover, the environment settings do not significantly impact the MI-based communication; this simplifies the development of communication protocols.

However, for many reasons MI-based solutions cannot be seen as a replacement for EM in WUSNs. Firstly, the bandwidth of MI communication is very limited (i.e., few KHz), which constrains its applicability in some scenarios. Secondly, recent theoretical results show that EM range can potentially be increased by more than 1 order of magnitude if the *lateral waves* effect is properly exploited in WUSNs [12–14]. Thirdly, MI-based communication cannot be directly realized between underground and aboveground nodes, thus creating challenges in the design. Finally, the nonexistence of commercial transceivers of this very recent technology (MI) also impacts its immediate adoption.

The ultimate conclusion given in this chapter is that both EM and MI technologies for WUSNs will experience significant development during the next few years.

Accordingly, we believe that cost-effective WUSNs will be heterogeneous networks with a mix of nodes using EM and MI technologies. In Sect. 3.2, the strategic use of both technologies is discussed and their complimentary features are highlighted.

1.1 WUSN Applications

While for some applications, the sensor itself needs to be buried (e.g., soil moisture probes can only function when buried among soil), it is not always clear why the transceiver/antenna also needs to be underground. One reason is concealment. For instance, due to aesthetic (e.g., in a golf field) or security (e.g., border control) concerns, some or all parts of a sensor node may need to be out of plain sight. In some other applications, wires and aboveground posts can potentially impact the regular activities. Moreover, the wiring infrastructure and electronic devices can be easily damaged by such activities, such as soil plowing, crop harvesting, etc. Two of such examples are precise irrigation (due to human and machine activity in the crop field) [8, 11] and sensors embedded in a sidewalk to provide guidance to the visually impaired [12].

A second reason is to prevent damage. For some applications the aboveground infrastructure such as wiring and posts may be damaged exactly when the network is needed the most (e.g., in landslide control). Finally, there are cases where the phenomenon being observed can only be captured by underground antennas (more detail is given in Sect. 2); that is, the antennas are the "sensors."

In [1] WUSN applications are organized into four classes: environmental monitoring, infrastructure monitoring, location determination, and security. We believe a fifth class, geophysical exploration, may be added. While similar to environmental monitoring, these are networks with much more specific needs as we explain below. There are two WUSN applications that have received increasing attention in the past few years: precision agriculture/farming (environmental monitoring) and oil leakage detection (infrastructure monitoring). The above classification is shown in Fig. 2 with each class explained in more detail next.

Environmental monitoring WUSNs are mostly associated with soil sensing with soil sensors buried underground. In particular, irrigation management is by far the

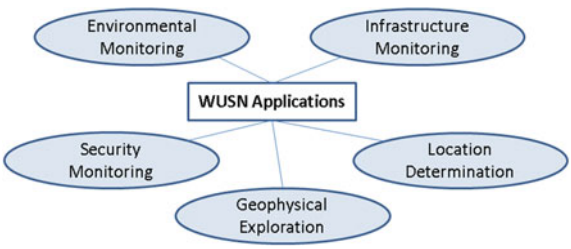


Fig. 2 Categories of WUSN applications

most important application in this class and also for WUSNs in general, in response to increased water shortage around the world.

Infrastructure monitoring is primarily related to underground infrastructures such as pipes and liquid reservoirs. The inspection of the foundation and internal structure of buildings, tunnels, bridges, etc., are potential extensions of this scenario. MI technology is mainly meant for this class of applications, as we will discuss in more detail in Sect. 3.

Location determination refers to the use of a network of underground nodes to forward location-related information to a mobile aboveground node passing through the area, which can be a robot, a car, or a person. An interesting application for the visually impaired using such a concept is suggested in [12].

Security monitoring underground sensor nodes naturally carry a much higher degree of concealment compared to other more traditional security solutions, and therefore this area is a potential one for the application of WUSN. For instance, even to detect the existence of such a WUSN can be extremely difficult; a *sniffer* device would have to be physically very close to the WUSN. One potential application is border-patrol [1, 15].

Geophysical exploration applications are associated with the potential use of underground communication as a *sensing technique* by itself. For instance, radar remote sensing (RRS) techniques have been widely used to study characteristics of *topsoil* (i.e., the top 30 cm of soil) and *subsoil* (usually the next 30–100 cm region) [16]. Depending on the frequency and soil makeup, the penetration depth of an RRS system can increase by an order of magnitude. Some geophysical investigations (e.g., oil prospection) require very deep measurements. A technique similar to RRS can be employed if a WUSN is formed at the boundaries of the area under investigation (deep soil). In this case, the wireless communication among nodes is used *collectively as a sensor instrument*. In other words, instead of using RSS systems, the characteristics of the medium under investigation can be inferred by the characteristics of the wireless communication between 2 or more nodes, in particular the signal attenuation. In this very particular class of WUSN application, the ultimate goal of the network is not the realization of wireless communication but the capture of information about the soil medium by means of the analysis of the signal attenuation.

1.2 Wireless Communication in Underground or Confined Areas

While our main interest in this chapter is WUSNs, it is worth mentioning a bigger class of wireless systems, those underground or in confined areas. Examples include systems deployed in mines, tunnels, and in certain disaster-relief scenarios. Different wireless communication technologies have been developed for such systems, and a taxonomy is given in Fig. 3. For a long time, the mining industry looked for effective wireless communication solutions to help trapped miners [12, 17]. Some of these options are categorized as Through-The-Earth (TTE) communication techniques. This is a point-to-point solution; more on TTE is discussed in Sect. 1.4. Wireless

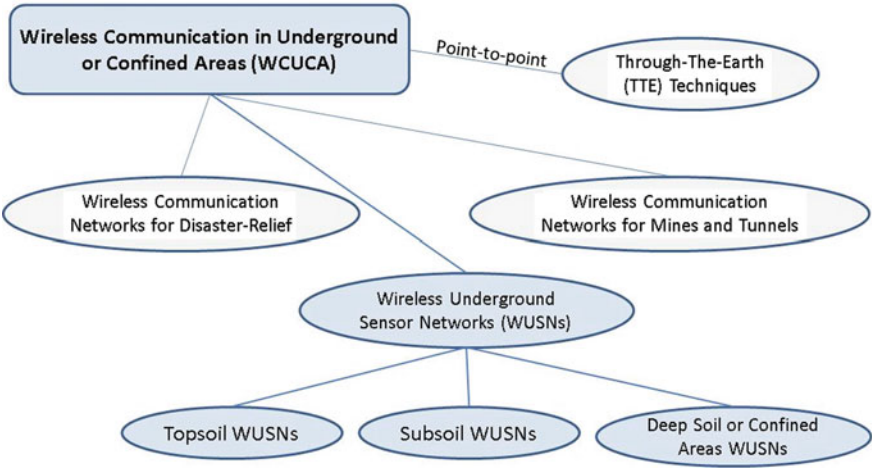


Fig. 3 Taxonomy of wireless communications in underground or confined areas (WCUCA)

communication techniques for *networks* located in underground and confined areas have been investigated in different contexts. Note that wireless networks used for mines and tunnels actually employ over-the-air communication.

We now turn to our main interest in this chapter, WUSNs. As shown in Fig. 3, such networks are classified as Topsoil (up to 30 cm depth), Subsoil (30–100 cm depth), and Deep Soil (>1 m depth) or Confined Areas WUSNs. Although the terms *topsoil* and *subsoil* typically match with the terms used in soil sciences [16], the exact length of each type of soil actually varies. However, such proposed classification for the WUSN context is primarily associated with two aspects: the physical security of the underground device and the signal attenuation level related to the underground communication. Devices buried at the Subsoil or Deep Soil are potentially protected against external agents (e.g., plowing activities in a farm, intruder, etc.). Second, EM-based communications in Deep Soil or in certain regions of Subsoil are potentially impacted if low-power transceivers are used, as discussed next.

Within this context, we define *burial depth* as the smallest distance between the antenna and the soil surface. Naturally, the closer the sensor node is to the surface, the easier is the installation and maintenance of that node. But the notion of burial depth has other significance as we discuss below.

The shallower the placement of a sensor node, the more achievable improvements in communication range. This is true for both EM and MI techniques, although for different reasons, as we will see in more details in Sects. 2 and 3, respectively. For an MI-based WUSN, it becomes economically feasible to install more and more MI-relay devices closer to surface. For an EM-based WUSN, the communication with aboveground nodes is greatly enhanced with shallower burial depths; as a result large and sparse networks involving underground nodes become feasible. Moreover, with shallower burial depths, the underground-to-underground communication range

also significantly increases due to the lateral waves (LW) effect at the vicinity of the soil–air interface. Therefore, the future of topsoil WUSNs (nodes buried up to 30 cm depth) is very promising.

Subsoil WUSNs are sometimes needed to protect nodes from potential damage due to aboveground activities, e.g., machinery activity in the crop field [8]. Note that even when the soil sensor is installed deeper, what really matters to the communication is the location of the antenna. Deep WUSN remains a relatively unexplored research area; a natural problem is the installation and deployment cost. To date, practical implementations based on EM waves are only feasible if high power transceivers (e.g., 2 W) are used. Moreover, deep burial depths limit the practical utilization of the LW effect because only the attenuation for the vertical parts of the signal path (i.e., from the transmitter up to the soil surface and again from the surface down to the receiver) can be high enough to impede the communication. By adopting a smaller frequency, such attenuation can be significantly reduced but the bigger dimensions of the antenna can be a problem considering the deployment requirements of some applications. For this reason, in this chapter, the term WUSN in the context of EM-based solutions will generally refer to topsoil and subsoil WUSNs considering the current available technology. On the other hand, MI-based solutions are typically applicable independent of the burial depth. MI technology is also a potential solution for WUSNs located at confined areas in general. However, typically MI modules involve higher physical volumes than EM modules; more about MI is discussed in Sect. 3.

1.3 Signal Propagation Through Soil

In this section, we introduce the main challenges behind any EM-based solution for WUSNs. When an EM wave propagates through a homogeneous dielectric material, the attenuation to the signal is governed by the frequency, distance, and conductivity/permittivity/permeability properties of the material [8]. However, the analysis becomes much more complex when a mixture of substances is involved. Soil typically consists of 3 components: sand, silt, and clay, each having distinct grain sizes as shown in Fig. 4. In this figure, the point P represents a soil type clay loam with 30 % of clay, 35 % of silt, and 35 % of sand. In addition, pockets of air may form in between these grains, which leads to sandy soil (the opposite is clay soil); the amount of air inside the soil is also a function of the soil density or how compact the soil is.

Besides sand, silt, clay, and air, soil also contains water, which is a very complex component to analyze for at least 2 reasons [18–21]. The first has to do with water dynamics in the soil: the environment and weather constantly affect the amount of water in the soil. Topography and soil makeup also influence the quantity of water in the soil, which is also referred to as the volumetric water content (VWC). Even locations within close proximity can have very distinct values of VWC.

The second reason why it is hard to analyze the effect of water on signal propagation through soil is related to the *type* of water present in the soil. *Bound water*

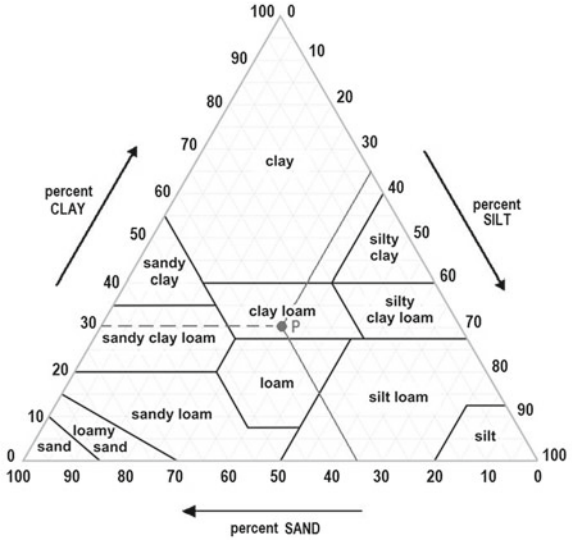


Fig. 4 Soil texture triangle based on the United States Department of Agriculture (USDA) classification [3, 20]

refers to water molecules that are tightly connected to the surface of the soil particles. By contrast, *free water* refers to water molecules that reside in the voids between soil particles. Due to the smaller size of a clay particle, clay soils typically have more bound water than sandy soils, and for this reason usually have higher VWC. If this does not sound intuitive enough, the following analogy may help. Imagine a box filled with 100 balls and we need to paint all of them. Consider now a second box with the same volume but filled with 10,000 much smaller balls. In the second case, a lot more paint would be needed due to the larger total surface area. In this analogy, the smaller balls correspond to the clay particles and the paint the bound water.

When the VWC is smaller, attenuation to the signal is also smaller. Thus sandy soil is typically more favorable to signal propagation. This, however, is not always true, a fact sometimes ignored when indoors experiments of WUSNs are performed [9]. Consider irrigating a sample of sandy soil and clay soil, respectively. Free water increases faster in the sandy soil than bound water does in the clay soil; thus during the irrigation process, sandy soil may pose higher attenuation. This is why during or immediately after a rain event, sandy soil can become extremely unfavorable for EM signal propagation. This example highlights the challenges in designing a sound communication system that can adapt to dynamic changes in the environment.

In addition, the VWC parameter alone is not sufficient for any EM attenuation model. The dielectric properties of bound water are different from free water; thus the attenuation to a signal as well as the change in its propagation speed (i.e., phase shift) vary as a function of both bound and free water amounts. Soil texture and its bulk density are also essential elements in the analysis.

If we ignore obstacles, the attenuation to a signal propagating over-the-air is typically modeled as a monotonic function of its frequency [22]; this is true for a large frequency band. This is also valid for many dielectric materials. Unfortunately,

this does not hold for soils. It is well known that signal attenuation in soil mixtures follows a very complex process. Depending on the frequency range, different models can be used to calculate the signal attenuation, a simple conductivity model or a more complex permittivity model. Let us assume that the operating frequency is at the 100 MHz–5 GHZ range.

The effective conductivity combines DC conductivity losses, polarization losses, and magnetization losses. Assuming a non-magnetic soil, the conductivity remains quite well-defined and stable for frequencies below certain values (e.g., 300 MHz), as empirically observed for different kinds of soil and water content levels [23]. This result can be explained by the fact that the excitation frequency is not close to the orientational polarization of free water and the polarization losses factor is small enough to be neglected. Therefore, for relative low frequencies, simple electric conductivity models maybe enough. However, as the frequency increases, the water polarization effect becomes more and more pronounced and the signal attenuation is better captured by a permittivity model that takes into account the amounts of free and bounded water at the soil. Geometrical characteristics of the soil particles are very important in this scenario and, as expected, the permittivity model becomes more complex.

In any case, if we limit our discussion to the 100 MHz–5 GHZ range, the soil attenuation will decrease if lower frequencies are used. However, the use of frequencies smaller than 300 MHz typically implies the adoption of non-practical antenna for WUSNs due to its physical dimension. Based on this fact, we can potentially narrow the EM frequencies for WUSNs to values above 300 MHz. At the UHF band (i.e., 300 MHz–3 GHz), knowledge of soil permittivity is essential, as previously explained. However, there is no simple way to obtain permittivity even if all parameters discussed so far are known. The difficulty is due to the impact of the geometric aspects associated with different soil makeups and density and how bound and free water are distributed in the soil structure.

Many studies have been done to establish a practical way to determine soil permittivity for different frequencies. Some of the proposed models are very accurate, but they require sophisticated instrumentation. Simpler approaches, usually semi-empirical models, can provide the answer for EM-based WUSNs. For instance, the Peplinski semi-empirical model [24] for the soil permittivity estimation was used in many WUSN works [3, 12, 25, 26]. Additional models are also referenced in the

Table 1 EM-based communication through soil: effect of soil properties on signal attenuation listed according to their impact (high to low)

Soil property	Effect on signal attenuation
Water content	More water: higher attenuation (critical)
Sand and clay composition	Clay soil: higher attenuation Sandy soil: smaller attenuation
Bulk density	High bulk density (soil is more compacted): higher attenuation
Temperature	High temperature: higher attenuation

WUSN works mentioned in this chapter. A summary of the effect of soil properties on signal attenuation is provided in Table 1.

The magnetic induction (MI) technique is not based on wave propagation, as we will see later in Sect. 3. MI is actually related to the variation of the induced magnetic field, similar to what occurs with the primary and secondary coils of a transformer. The soil is actually the core of this "transformer" system. Because, the soil is typically a non-magnetic medium, the signal losses exist and still are significant. In fact, the non-propagating magnetic field (quasi-static) decay is roughly governed by an inverse cube law and the communication range of the system is also impacted. However, the key difference in relation to the EM losses discussed in this section is the fact that the magnetic field losses are not strongly influenced by the soil type and especially by the water content. Moreover, the multi-path interferences will not occur for the MI scenario.

1.4 TTE Techniques and WUSNs: A Comparison

Through-The-Earth (TTE) solutions provide some form of wireless emergency communication to trapped miners in a disaster scenario. Usually, a point-to-point approach is sufficient for this purpose. On the surface one might expect to find many similarities between TTE and WUSN technologies. Moreover, because TTE techniques have been in use for a long time, one might also expect to reuse many of the TTE techniques in WUSNs. Unfortunately, this is not the case because the characteristics of typical TTE and WUSN solutions are significantly different, as shown in Table 2.

One of the most important differences between TTE and WUSN is the communication range. Usually, a powerful TTE device is placed at the soil surface. This equipment is used to transmit data many hundreds of meters down in the direction of possible locations where the miners are. This aboveground device can transmit

Table 2 Comparison between Through-The-Earth (TTE) techniques and WUSNs [12]

Design aspect	TTE-based communication	EM-based WUSN
Frequency band	VLF/LF	VHF/UFH
Maximum range (soil path)	Up to hundred meters	Up to dozen meters
Bandwidth	Very small: bps	Small: Kbps
Network topology	One-hop	One-hop or multi-hop
Network size	Sender-receiver pair or few nodes	Up to hundred of nodes
Underground channel noise	Very critical	Small impact
Rock penetration	Feasible	Typically not possible
Soil moisture impact	Small impact	Very critical
Energy criticality	Relatively small impact	Very critical
Node cost	Relatively high	Small
Communication protocol design	Emphasis on the physical layer	Cross-layer approach

a strong signal without critical energy/processing constraints. On the other hand, energy-constrained TTE devices located in the mine can severely limit the bottom-up communication performance. TTE communication has been used for 3 classes of applications [12, 17]: miner locating systems, geophysical exploration, and military underground communication during the nuclear age. Currently, TTE devices are mainly used for the first class of applications. However, due to recent developments in WUSNs, especially in the MI area, it is conceivable that TTE solutions will eventually adopt more and more WUSN technologies.

2 Electromagnetic-Based Solutions (Radiation Field)

In this section, the most recent WUSN development based on the propagation of Electromagnetic-Based (EM) waves at the UHF band are examined. Note that the signal propagation path is not limited to the underground settings. For instance, some designs require no communication between the underground nodes; they always communicate with an aboveground node [1, 10, 11]. More elaborated solutions, also called *Hybrid WUSNs* [1], combine different modes of communication at the same time. To better distinguish different types of communication links, we give the following classification, also shown in Fig. 5:

- **Underground-to-underground (UG2UG) Link**: the communication between the sender and receiver occurs without the assistance of an aboveground node. Typically, the propagation path is entirely in the soil. For some scenarios, when the LW effect is considered, part of the path is over-the-air.
- **Underground-to-aboveground (UG2AG) Link**: the receiver node is above the ground and the sender is buried. This link accounts for the most common need for WUSNs, i.e., sensing data on the underground environment is transmitted to aboveground relays or sinks.
- **Aboveground-to-underground (AG2UG) Link**: the sender node is above the ground and the receiver is buried. This link is typically used for management purposes, such as informing the underground node about measurement schedules.

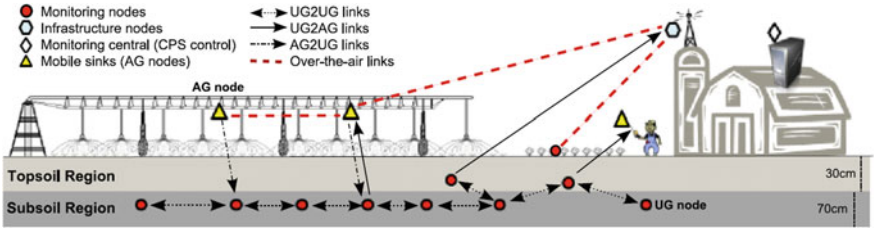


Fig. 5 Typical Topsoil and Subsoil WUSNs employ 3 types of communication links: underground-to-underground (UG2UG), underground-to-aboveground (UG2AG), and aboveground-to-underground (AG2UG) [11, 12]

Empirical results show that the above 3 links have very distinct characteristics and a single communication model is not possible [8, 10, 11]. Therefore, we will analyze each separately. We will begin with an introduction to the buried antenna problem, as all 3 WUSN links have at least part of the wave propagation path through soil.

2.1 Buried Antenna Problem

The first studies related to the EM waves propagating in a medium different from air/vacuum began in 1909, with the work of Sommerfeld analyzing the radiation of an electric Hertzian dipole in a dissipative half-space [27]. Although these studies consider dipole sources located in free-space (the antennas are just above the soil surface), the generic free-space Friis equation does not apply in this scenario. The pioneer investigation of the radiation of a Hertzian dipole immersed in a conducting medium, such as the soil, is realized by Tai in 1947 [28, 29]. Since then this has been a subject of intensive study.

The classic work of Banos in 1966 [30] includes a complete characterization of the electromagnetic field components for points in the dissipative medium (sea water) or above it (air). At that time, it was clear that the radiation model cannot be achieved by a closed formula. However, approximate formulas can be achieved for certain frequency bands and specific media. In fact, the Banos formulas are only valid for a conducting medium, such as sea water, and at low frequencies.

The *antenna problem* is not included in these pioneer works because only ideal elementary source dipoles were considered. However, in order to fully characterize the wireless underground communication, two problems must be simultaneously solved: the propagation problem, as previously mentioned, and the antenna problem. More recently, in 1980, King [23] partially addressed the antenna problem for lossy medium such as soil. It is observed that, for underground communication purposes at UHF frequencies, only *insulated* antennas can be considered [23, 29]. In fact, in the experiments reported in [8, 9], one failure in the antenna insulation invalidated a set of experiments due to the resulting additional attenuation.

Only some important classes of embedded insulated antennas are considered in [23]: dipole, loop, and terminated monopole (travelling wave antenna). For all these cases, the subsurface is the soil region of choice (for very high power solutions, deeper installations are also feasible). The reason behind this constraint is the LW effect. The mathematical and physical justifications for the existence of lateral waves and their use in radio communication are provided in the classic work of Brekhovskikh [31].

Buried/immersed antennas and lateral waves propagation were mainly studied on the 1940–1980s due to two main application scenarios at that time: (a) communication with submarines and (b) protection of the communication system of a country in the case of a nuclear attack. Since then, a lack of potential applications for LW likely explains the relatively low activity in this area during the last few decades. The continuation of these original studies by extending and adapting them to the WUSN scenario is first proposed in [12] and such research was extended in [13, 14].

When an antenna is immersed in a dielectric medium, its design must be adjusted. The typical design found in the literature considers a non-isolated antenna in air/vacuum. In this case, the wavelength is calculated considering the light speed. In soil, wave propagates in a lower speed resulting in a smaller wavelength for the antenna design. However, as explained in Sect. 1.3, changes in the soil characteristics and VWC modify the dielectric constant of the soil surrounding the antenna over time, resulting in changing wavelengths. Therefore, the design of an antenna must take this dynamic aspect into account. An ultra-wide band antenna can be an option, as demonstrated in [4–6, 10–12].

Adjusting the wavelength is just one of the design aspects of the buried antenna. Its radiation pattern must also be carefully evaluated according to the locations of potential neighbors of a node (i.e., directivity). One aspect common to all WUSN studies done so far is that the communication range between two devices is drastically reduced compared to the over-the-air case. Therefore, the 1-hop neighborhood of a node is also significantly smaller. Moreover, if the LW effect is exploited, the best approach is to use an antenna with high directionality toward the area above the buried antenna.

2.2 EM-Based Underground-to-Underground Communication

In this section, the state-of-the-art EM-based UG2UG communication for WUSNs is discussed with a focus on the physical layer. We start by analyzing the first proposed UG2UG model, which is a variant of the well-known two-ray model [22] for the underground setting. This model considers two components of the received signal: the direct and reflected waves (DW and RW) as shown in Fig. 6. Related work up to 1980 was briefly mentioned in the previous section. Additional studies, some of them related but not directly targeting WUSNs, are presented in [1]. An important research area that provides the foundation for the current wireless underground communication models is microwave remote sensing [18, 19, 22, 32, 33]. Similarly,

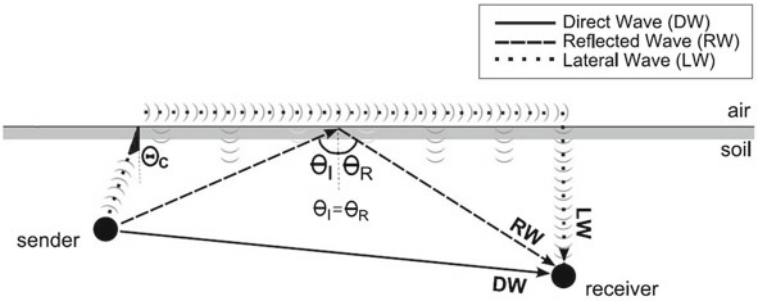


Fig. 6 The received signal is a superposition of direct waves (*DW*), reflected waves (*RW*), and lateral waves (*LW*) [12–14]

studies on detecting landmines using Ground Penetrating Radar (GPR) [34, 35] also help us understand why multiple soil permittivity models exist.

An overview of the challenges related to the WUSNs is provided in [1]. The challenges for performing outdoor WUSN experiments are discussed and guidelines are provided in [9]. A theoretical model specifically for wireless UG2UG communication is proposed in [1], though without empirical results to validate the model. More detailed information of this model is also reported in [26].

UG2UG experiments using Mica2 motes at 433 MHz are reported in [8] and the results show a good agreement with the model proposed in [26] for high burial depths. Unfortunately, even considering the best soil conditions and very small burial depths, the estimated inter-node distance is smaller than 5 m. For deeper deployments (e.g., 1 m), the inter-node distance is smaller than 1m. Although such results are insufficient in general for WUSNs, they do enable some applications. For instance, in [36], a very small inter-node distance of 30 cm is enough for that WUSN application.

The mismatches between the model in [26] and the empirical results for low burial depths suggest the existence of a missing factor not considered in that model. This factor is the lateral waves (LW), as pointed out in [12] where a more complete model is proposed; this is shown in Fig. 6. It became clear then that medium to long-range UG2UG communication can only be realistically achieved if the system is specifically designed to take advantage of the LW effect. Alternatively, a very high density of nodes and/or the use of very high power transceivers can be employed. We believe that the LW effect is the key to successful WUSN designs based on EM waves. However, it is important to highlight that this statement is particularly significant for shallower burial depths (topsoil and some subsoil WUSNs). For instance, if two typical 433 MHz WSN nodes are installed at 1m-depth (deep soil), only the attenuation due to the up-down path of the signal in fair soil conditions can be enough to impact the communication (e.g., >100 dB [8]) and the advantages of the LW are impacted in this context. Although smaller frequencies can still be used to minimize the mentioned signal attenuation, there are tradeoffs in relation to practical antenna sizes that must be taken into account.

More recently, the role of LWs in the UG2UG communication has been highlighted in the literature and two UG2UG models containing the LW component have been proposed [13, 14]. However, empirical UG2UG work is still limited to short inter-node distances (e.g., <3 m) and the proposed models that consider the LW component still requires more significant empirical validation.

We next provide a more detailed historical survey on UG2UG applied to WUSNs.

In 2006, commodity sensors MicaZ motes (2.4 GHz, 0 dBm transmit power level) were tested for UG2AG and UG2UG links in [37]. The UG2UG communication was reported infeasible for that specific testbed scenario. In 2009, many aspects related to the UG2UG experiments in outdoors were highlighted in [9]. That year new UG2UG experiments were realized using very well-controlled procedures [8]. MicaZ and IRIS motes (2.4 GHz, 0 and 10 dBm transmit power, respectively) were tested and successful UG2UG communication was achieved for inter-node distances of up 20 cm for a typical soil condition. Although such distance is still useless for many practical WUSN implementations, this result indicates that UG2AG and AG2UG

communication is potentially feasible when the underground node is installed at shallower depths. As discussed in subsequent sections, this was confirmed by both theory and practice.

In 2008, a study was published on WUSN experiments based on UG2AG and UG2UG communication and a customized 27 MHz mote and 30 dBm of transmit power [38]. As expected, the lower frequency allowed a significant increase in the inter-node distance. In this case, a distance of 20 m (17 % frame error rate) was reported on the UG2UG link with 30 cm-burial depth. Nonetheless, the power consumption and the antenna size (2.8 m) were still limiting factors for many WUSN applications.

In 2009, outdoor experiments using Mica2 motes at 433 MHz confirmed that low frequencies at the VHF band are specially promising for WUSNs due to performance and antenna size considerations [8]. Inter-node distances of 50 and 90 cm were achieved with 0 and 10 dBm of transmit power (40 cm-burial depth), respectively. Theoretically, in sandy soil with smaller water content the distance may reach 1–2 m using the models in [12–14, 26]. Moreover, the same work revealed that a significant improvement was achieved when shallower depths were used. By then reported WUSN work was mainly based on UG2AG/AG2UG communication.

An additional feature of the UG2UG communication channel revealed in [8] was that it apparently was highly stable: the overall packet error rate (PER) over a 24h-period was smaller than 0.5 % with a very small variance. In other words, once communication is achieved even under a high soil moisture condition, the probability of error is very small. This phenomenon is explained by the fact that noise encountered over-the-air is drastically reduced in the underground domain. This observation points to at least two design strategies. First, error control can be greatly simplified or even eliminated, minimizing communication overhead. Second, it is possible to design UG2UG communication for low VWC (thus, low power consumption) and temporarily suspend network operation when the VWC drastically increases (e.g., during rainfall). Note, however, for the second strategy to work, the sensor node must be equipped with a moisture sensor.

In 2010, guidelines for the development of an outdoor WUSN testbed were proposed in [9]. The goal was to improve accuracy and efficiency in conducting WUSN experiments. In particular, a typical problem encountered outdoors was mentioned: since sophisticated equipment is rarely used, especially if such equipment must be buried, the common approach of using radio to take signal strength measurements suffers from what is called *clipping effect*, when the received signal strength (RSS) reported by the radio is smaller than the actual value.

Also in 2010, the feasibility of an UG2UG link on the seabed (100 m below the sea) was investigated in [39]. This study is especially important when deep WUSNs are considered for geophysical exploration. Simulations were done for different frequencies and a high transmit power (40 dBm = 10 W). Besides the expected high inter-node distance, the seabed scenario presents a very high VWC, far beyond the typical values for soil. As expected, VHF frequencies such as 700 MHz are associated with very high path loss: 12 and 40 dB per meter for 5 and 50 % VWC, respectively. On the other hand, for 1 MHz the path loss was reduced to 1 and 3 dB per meter,

respectively. The conclusion of this study was that low frequency combined with a multi-hop topology (with nodes acting as relays) is the proper solution for that scenario.

In the above case, the antenna size was not a critical consideration. However, for the majority of WUSNs big antennas are associated with higher deployment complexity and cost. Acceptable antenna sizes limit us to the VHF band; the resulting inter-node distance for UG2UG (ranging from less than 1 m to a few meters) is likely far from ideal for many WUSN applications, though it is sufficient for some application scenarios. Two such examples are [40, 41]. In [40], a WUSN was proposed for the detection of soil displacement during the excavation of a gallery, and the UG2UG link was reported to be feasible. The "soil" medium in this case is gravel inside a metallic tube and the prototype of the system is considered successful for an inter-node distance of up to 1m. This is a rare example of a Deep Soil WUSN, as defined in Sect. 1.2. In this case, the metallic pipe potentially acts as a waveguide. A second example is given in [41]. In this study, a sensor node is embedded in wet concrete and used to continuously monitor concrete curing and structural health. At 433 MHz, the maximum inter-node distance (in terms of concrete thickness) was measured to be 50 cm. In all, the receiver can be up to 5 m away from the concrete structure: 4.5 m over-the-air and 0.5 m through wet/dry concrete. Moreover, this study showed that while the concrete is curing (high humidity), the communication performance is deteriorated, similar to the effects of soil moisture reported in [8, 26]. Again, the small inter-node distance of EM-based UG2UG links was not a limiting factor in this scenario.

In 2011, an UG2UG model that highlights the importance of the electrical conductivity of the soil was proposed [42] and empirical results were presented. For the first time, it was shown that 2.4 GHz commodity motes (MicaZ) can be used to realize UG2UG communication. A PVC box was buried at 1.4 m-depth with multiples nodes immersed in different kinds of sand and soil moisture conditions. Under these conditions, the maximum inter-node distance of approximately 1m was achieved. In the proposed model, only direct waves was considered (no RW or LW waves), making it suitable for deep WUSNs.

The majority of the WUSN implementations require a larger inter-node distance, e.g., 5–50 m. Solutions to this challenge come in two flavors: LW effect for EM-based solutions and magnetic induction (MI). In 2010, as previously mentioned, for the first time the LW effects were linked to WUSN in [12]. In 2011, three additional UG2UG models were proposed. The first one [43] is similar to the one proposed in [26], also without considering the LW effects, and simulated results were presented. The second proposed model [13] includes the LW effect for subsoil and in particular for topsoil WUSNs. In the third work [14], an UG2UG model based on [26] was extended to include the LW component. It was shown via simulation [12–14] that a commodity WSN mote can achieve successful UG2UG communication over higher distances, such as 5–10 m. In addition, the work in [14] considers the network connectivity aspect for WUSNs that involve UG2UG, UG2AG, and AG2UG links.

Despite rapid advances in modeling the UG2UG communication channel, the models presented so far do not fully characterize the problem when the "antenna

problem" (Sect. 2.1) is not taken into account. This is because antennas in these models are assumed to have the same behavior as in free space. More specifically, the effects of the directivity and other performance parameters of the antennas are simplified by the introduction of a *fixed* term in the antenna gain. While this practice works well for over-the-air scenario, it is no longer appropriate for buried antennas. In [12], the antenna factor was included in the model as a *dynamic* parameter to be empirically determined given a set of parameters related to the deployment and environment. Still, more comprehensive studies showing the best antenna design practices for a given WUSN scenario are currently lacking. For instance, if the LW effect is to be enhanced in order to achieve a higher inter-node distance, Silva [12] suggested that a directional antenna with a sharp beam pointing in the direction of soil interface and with an angle close to the critical angle is the best approach. However, this approach is only part of the solution. In 1963, extensive experiments were done with antennas buried at different depths [44]. It was observed that the radiation pattern of the antennas is significantly disturbed when they are close to soil interface. Part of the explanation has to do with stronger presence of reflected and lateral waves. Also, as already explained, the antenna was designed for over-the-air wavelength and the change of wavelength in underground settings affects the antenna. Moreover, the antenna impedance also changes with a number of parameters, such as soil moisture, potentially causing an additional loss. The current set of models does not fully address the antenna factor, at least not in an analytical form.

We end this section with a more detailed discussion on lateral waves and the antenna problem.

As already explained, the LW effect should be exploited in order to achieve inter-node distances that are desirable for a broader class of WUSN applications. The theory of LW is addressed with a rigorous mathematical approach in [23, 31, 45]. Below we give a qualitative explanation of this phenomenon applied specifically to WUSNs.

Consider the interface of two materials with different dielectric constants: a thinner medium, such as air, and a denser medium (smaller propagation speed, higher permittivity), such as soil. Assume that both sender and receiver nodes are at the denser medium but close to the interface. When the EM waves reach the interface precisely at the critical angle by Snell's law, they are refracted at the thinner medium (e.g., air) and propagate *along* the interface while continuously penetrating the denser medium (e.g., soil) at the same critical angle. There is no geometric-optical representation for this phenomenon, but it is not hard to realize that the energy associated with the wave front at the critical angle does not disappear. The LWs never radiate through the thinner medium, but return to the denser medium as shown in Fig. 6. For simplification, we usually represent only the wave front (graphically as a ray) that reaches the point of interest (where the receiver is located). This explains why this propagation mode is sometimes called "up-lateral-down." Note that we cannot simply assume that the received power is the transmit power subtracted by the dense medium path losses ("up" and "down" paths) and by the thinner propagation loss (e.g., air). In fact, the received power is much smaller. The solution to this problem does not have a closed form and usually approximations are used [12–14, 23, 27–31, 45].

It is worth noting that the LW effect is rarely mentioned in communication system textbooks. This is because usually the sender and receiver nodes are not located in the vicinity of an interface as described above. Also, even when the interface is present, the medium where the sender and receiver are located is typically not as lossy as the soil. Therefore, the LW effect on the received signal is negligible compared to direct or reflected waves.

By contrast, for typical soil conditions and inter-node distances higher than 3m, the tiny contribution of LWs becomes the dominant factor in the received signal strength. Therefore, communication over high distances is only possible if LWs reach the receiver with sufficient energy, and thus design of the UG2UG link must take into account the LW effect when shallower depth deployment is the case. Fortunately, there is a way to improve the communication range without increasing the transmit power, through the use of directional antenna. Specifically, only a tiny fraction of the transmitted waves from an isotropic antenna impinges the interface at the critical angle, resulting in huge energy waste (dissipated into the soil or radiated into the air). On the other hand, if the antennas on both sides have a sharp beam (pencil-like) in the upward direction and centered at the critical angle, the results are significantly improved due to the higher LW contribution. It should be noted, however, that variations of soil parameters such as soil moisture can change the critical angle, and that antennas with very high directivity can be hard to build.

2.3 Underground-to-Aboveground and Aboveground-to-Underground Links

In this section, the state-of-the-art on UG2AG and AG2UG communication is presented. MI-based techniques may not be the proper solution for typical WUSN scenarios because the signal attenuation due to MI technique for an over-the-air communication path is significantly higher compared to the EM technique. Nonetheless, over-the-air communication using MI are mentioned in [46, 47]. However, considering the use of low-power transceivers and the distances involved in the WUSN scenarios presented in this section, only EM-based solutions are considered. Differently from UG2UG, the communication range in this new scenario can be extended to relatively long distances (e.g., 30 m) with low-power transceivers. Since higher burial depth means worse performance, these communication links are only feasible for subsoil and especially topsoil WUSNs.

Some commercial products are available in this context [48, 49] and, frequently, a star-topology is adopted. The underground nodes typically communicate only with aboveground nodes in the vicinity. Multi-hop communication among aboveground nodes (over-the-air) eventually provides full connectivity for the network, including the sink node(s). This is a simple but very effective approach to WUSNs. In fact, wherever possible, the use of aboveground nodes greatly reduces the complexity and increases the reliability of WUSNs. Moreover, mobile aboveground nodes have been suggested as a way to increase the physical coverage of a highly sparse WUSN [1].

However, the realization of UG2AG and AG2UG communication is not without challenges. Depending on the soil makeup and condition, the communication range can significantly differ under otherwise identical conditions. Asymmetry is another prominent feature: a successful UG2AG communication does not imply that AG2UG will succeed and vice-versa [10, 11]. Moreover, for the UG2AG link, the radiation pattern of the buried sender node (with an omnidirectional antenna) in its vicinity is highly irregular. We have mentioned this phenomenon in the UG2UG context, and it was empirically observed for different frequencies, environmental, and test conditions [8, 44]. Moreover, the antenna is a critical design factor for UG2AG and cannot be simply assumed as represented by a static gain. Similarly, the same comments apply for AG2UG links.

Below we provide a survey on UG2AG and AG2UG communication, again with a focus on the physical layer. The research in WUSN actually started with UG2AG communication as it corresponds to the primary goal of an underground node: to collect data from the underground environment. In 2006, commercial sensors MicaZ motes (2.4 GHz, 0 dBm transmit power level) were tested and UG2AG performance presented in [37]. Two sets of tests were conducted: the receiver node at the soil surface and elevated 1m above soil surface. The maximum horizontal inter-node distances achieved, with packet error rate less than 10 %, were 2.5 and 7 m for 13 cm and 6 cm-burial depths, respectively. Very similar results were achieved latter (2009) using TelosB mote (2.4 GHz and 0 dBm of power level) [50].

Also in 2006, a wideband antenna for UG2AG communication was proposed [5]. As discussed in Sect. 2.1, when the antenna is buried the target wavelength in antenna design is smaller compared to that in the over-the-air case. However, because soil moisture has temporal and spatial variation, antenna design must take into account a wide range of wavelengths. The proposed wideband antenna mitigates this issue. The same author subsequently proposed a communication model for the UG2AG link [6]. In this study, the communication was unidirectional and the AG2UG link was absent. A simple star-topology was adopted. A customized sensor node called Soil Scout (868 MHz, 10 dBm of transmit power) was used with the wideband antenna. The results showed a radiation efficiency of more than 90 % for the proposed antenna in different soil textures and VWC levels. A special high-gain antenna, elevated at 12 m, is used at the receiver. Communication ranges of 30 and 150 m were reported for burial depths of 40 and 25 cm, respectively. This UG2AG model assumes long-range links (e.g., >20 m) while communication in the vicinity of the sender was not considered.

In 2007, a real-world deployment of a WUSN with both UG2AG and AG2UG on a golf field was reported in [51]. 18 customized nodes (868 MHz, 10 dBm) were buried at a depth of 10 cm (approximate) and 24 aboveground nodes at the same frequency form a mesh network. The inter-node (UG2AG) distance was reported to be 62 m.

In 2008, an empirical model for the UG2AG link for a water distribution monitoring system was proposed [52]. In this case, a signal generator (2.4 GHz, 24 dBm) was placed in an air-filled underground fire hydrant (concrete box), while the aboveground node was a spectrum analyzer with an antenna placed at different heights.

There was no soil above the sender's antenna, but elements including the concrete, the soil *outside* the fire hydrant, and its metallic lid were factors that could degrade the communication performance. The maximum and minimum horizontal distances achieved for reliable communication were determined to be 38 m (north direction/lid on/4 m-height) and 4 m (north direction/lid off/3 m-height or south direction/lid on/2 m-height), respectively. The impact of soil makeup and condition in the area surrounding the fire hydrant was not investigated in this work.

In 2009, a unidirectional UG2AG communication model was proposed [7] and simulated and empirical (laboratory) results were presented. A customized sensor node SoilNet (2.4 GHz, 19 dBm of transmit power) was used for the experiments. The burial depth for the device is 9 cm (topsoil WUSN) allowing the model to be simplified. It concluded that an inter-node distance of up to 400 m would be possible. A real-world implementation based on SoilNet devices was presented in [53], and inter-node distances slightly smaller than 100 m were successfully achieved (based on the sensor map given in the paper). The AG2UG link was also shown to work properly, although an explicit model for this link was not provided.

The previous work [6, 7, 51] demonstrates that long-range communication in WUSNs is realistic with *shallower* deployments. Commercial solutions following this type of network topology (aboveground node at the center of a star-topology) are already available. A residential irrigation system for shallower depths (10 cm) has a nominal inter-node distance of 180 m [48]. Another similar solution at 900 MHz and the same burial depth has a range of 152 m [49].

Also in 2009, well-controlled UG2AG experiments were conducted using commercial Mica2 motes (433 MHz, 10 dBm) with their original monopole antennas [54]. One of the goals of the experiment was to observe in low-power devices the radiation pattern of a buried antenna. Previous work (in 1963) [44] had concluded that a typical over-the-air radiation pattern is significantly disturbed when the antenna is buried. Throughout the test, the soil was carefully examined in order to remove any kind of obstacle (rocks, plants, etc.). Moreover, the tests were performed at 2 distinct sites with homogeneous soil makeup and without surface irregularities. As shown in Fig. 7, the typical radiation pattern of the original monopole antenna was significantly modified underground. For instance, observe that the packet error rate (PER)

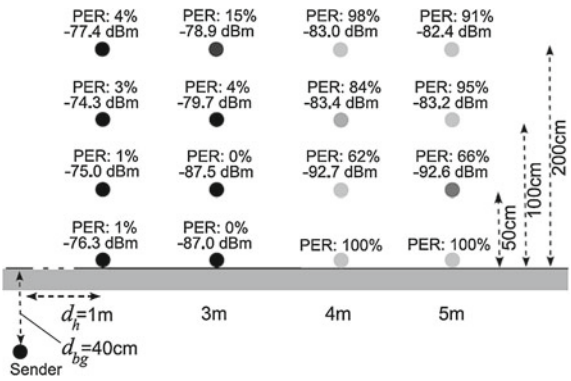


Fig. 7 The radiation pattern of an underground node is not typical; it does not follow its over-the-air counterpart [54]

at $d_h = 3$ m and 50 cm-height is 0 % with a signal strength of −87.5 dBm (acceptable for Mica2). However, at $d_h = 4$ m and 100 cm-height, the signal strength is higher (−83.4dBm), but the communication is practically infeasible (PER = 84 %). In the same paper, a second experiment (different site) presented similarly unusual behavior. Moreover, such irregular radiation pattern changes with soil moisture. This result highlights the importance of considering the antenna factor as a *dynamic* component in any theoretical model for WUSNs, as pointed out in Sect. 2.1. However, based on the results of the majority of UG2AG/AG2UG works previously discussed, this problem does not seem to be significant for shallower burial depth and/or long-range (i.e., far-field region) links.

In the same year, field experiments for the UG2AG and AG2UG links were done with Mica2 mote (433 MHz, 10 dBm) with aboveground antenna height fixed at 2.5 m [10]. Different combinations of sender and receiver antennas were tested, starting with the motes' original monopole antennas. With an ultra wideband antenna for the underground node (35 cm-burial depth), the UG2AG horizontal distance was enhanced by more than 266 % and the AG2UG was realized (13 m) at that depth. When the burial depth changed from 35 to 15 cm, the communication range is increased by 40 and 300 % for UG2AG and AG2UG links, respectively. Moreover, a 21 % increase in soil moisture led to more than 70 % decrease in communication range. It also reported for the first time that UG2AG and AG2UG links are asymmetric, though typically there is a region where both enjoy good performance and thus symmetry between the two may be assumed. This complex scenario naturally requires strategic WUSN designs. Usually, UG2AG links have better communication range performance. If we return to Fig. 5, we can observe an interesting strategy: the sink at the top of the building is properly collecting data from some buried nodes. However, if control data (such as configuration) must be sent to an UG node, another AG node closer to this one can be used as a relay node.

The first real-world WUSN experiment involving a mobile sink and buried nodes was demonstrated in [11]. It was done in a corn crop with a center-pivot irrigation system. The aboveground node (sink) was fixed at the mobile irrigation system and 8 nodes were buried at the root zone (around 40 cm). All nodes were Mica2 motes (433 MHz, 10dBm). The goal was to determine if it was possible to automate the system and control the amount of water based on sensor information. The average horizontal distance before the center pivot reached a certain area was found to be 6m, proving the feasibility of the proposed solution.

In 2011, a probabilistic routing protocol for UG2AG/AG2UG links was proposed in [55] with an outdoor testbed composed of 17 TelosB motes (2.4 GHz, 0 dBm transmit power level) in a 20 m × 20 m area. Different burial depths were analyzed: 0, 10, and 20 cm. In addition to empirical results, simulations were also presented and discussed.

In 2012, a WUSN architecture is designed consisting of mobile nodes that harvest data from stationary underground nodes. In this work, the impacts of packet size and error control schemes (ARQ and FEC) on network performance are investigated and a family of WUSN protocols is proposed. Empirical experiments based on Mica2 motes

are realized and the advantages of having mobile nodes in WUSNs are highlighted in this work.

In the same year, it is proposed a compressing technique for in-situ soil moisture measurements [56]. Although, it is an application-related effort, it reveals the fact that many WUSN applications related to soil parameters measurements can employ techniques to minimize the amount of data flowing in the network.

In 2012, a cross-layer solution for static and low duty-cycles WSNs [57] is proposed in order to achieve very high energy efficiency for such scenarios. Although the presented experiments only involve aboveground WSN nodes, the solution proposed in this work is specifically designed considering nodes with non-rechargeable batteries in large and sparse networks. Such aspects can potentially match the deployment requirements of many WUSNs

As a conclusion for this section, the main challenge for EM-based solutions to WUSNs is propagation loss in the soil that also depends on the soil composition, density, and moisture. Such loss can vary from 20 dB/m to more than 100 dB/m in various scenarios. Assuming a dynamic range of almost 100 dB for the radio transceivers (typical), the inter-node distance between 2 buried sensors can vary from a few centimeters to 5 m in the best case (assuming typical commodity nodes used in WSNs). Although reflected waves from the soil surface are associated with a positive effect [8, 26], their contribution is not significant. Two strategies can mitigate the problem: shallower burial depths and LW-based solutions. The first, shown to be successful in UG2AG/AG2UG communication, achieves lower soil path attenuation simply due to shortened path length. However, not all applications can allow shallower deployments, precision agriculture being a prime example [8, 11]. The second strategy (LW) deserves attention especially because to date it has not been fully exploited. If antennas are properly designed to enhance the LW effect, the UG2UG inter-node distance can increase significantly, theoretically going beyond 10 m [12–14]. To achieve even higher distances in practice, such as 50 m, MI-based solutions emerge as the preferred option, although only very low bandwidth can be achieved using the state-of-the-art MI technology, as we will see in Sect. 3.

In this section, the state-of-the-art EM-based UG2UG communication for WUSNs is discussed with a focus on the physical layer. We start by analyzing the first proposed UG2UG model, which is a variant of the well-known two-ray model [22] for the underground setting. This model considers two components of the received signal: the direct and reflected waves (DW and RW) as shown in Fig. 6. Related work up to 1980 was briefly mentioned in the previous section. Additional studies, some of them related but not directly targeting WUSNs, are presented in [1]. An important research area that provides the foundation for the current wireless underground communication models is microwave remote sensing [18, 19, 22, 32, 33]. Similarly, studies on detecting landmines using Ground Penetrating Radar (GPR) [34, 35] also help us understand why multiple soil permittivity models exist.

3 Magnetic Induction-Based Solutions (Static Magnetic Field)

Magnetic induction (MI) does not significantly suffer from *soil path propagation loss* compared to EM-based solutions simply because the communication is not based on wave propagation. Although the transmission also involves alternating electric and magnetic fields, the combination of low power and low frequency causes the radiation effect to be extremely minimized. Naturally, path loss still exists especially that related to magnetic coupling.

MI techniques applied to WUSNs works in a way similar to the step-down lines transformers used in power supplies years ago. Radiation exists in such devices, but is mainly limited to the vicinity of the transformer. Nonetheless, the effect of the magnetic induction on the secondary coil of the transformer is responsible for transferring majority of the energy. If we turn on/off the alternating current at the primary coil, such effect is "communicated" to the secondary coil of the transformer. This might help us understand how it is possible to achieve communication using MI while neglecting the radiation (i.e., propagation) effect. Essentially, the radiation phenomenon is not of interest in this case, so the electric field of MI transceivers can be minimized or eliminated. Only the magnetic field is effectively used and the communication region of interest is the static field (also, the quasi-static region) in contrast to the far-field region typically used in communications based on the propagation of EM waves.

Below we start with a survey on MI again with a focus on the physical layer. Then in Sect. 3.2, we give a comparison of EM and MI techniques applied to UG2UG links in WUSNs and discuss hybrid solutions.

3.1 Related Work

In 1997, unidirectional and high-power MI communication at 3KHz was investigated for military operations in coastal regions [58]. Successful results were reported at data rates of up to 300 bps. This work anticipated what would be a constant in MI communication: relative low frequencies and very small bandwidth. In 2009, the same company behind the work in [58] commercialized a relative low-power device, MI Remote Activation Munition System [46], which achieved communication range of 200 m through soil, rock, vegetation, and water. It also commercialized TTE devices (discussed in Sect. 1.4) based on MI technology.

In 2001, the advantages of using MI for wireless close-proximity (e.g., <3 m) applications, such as those using Bluetooth devices, were highlighted in [59]. A comparison with high-frequency RF systems was given and smaller power consumption and complexity were singled out as main features of MI-based solutions. It also pointed to 11–15 MHz carrier for MI communication systems. Some of the current proposals for WUSNs operate at this frequency range.

In 2002, the theoretical model of a simple electric circuitry for MI devices was discussed in [60]. This circuitry would eventually become the basic cell in MI

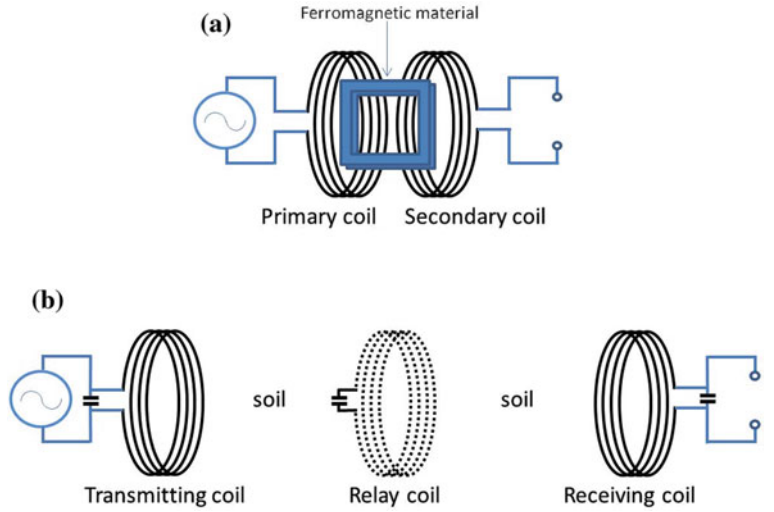


Fig. 8 **a** Ordinary transformer. **b** MI system based on resonant LC circuits with optional use of additional relay coil(s) (MI waveguide effect)

communication for the ensuing years. As shown in Fig. 8, both transmitting and receiving coils have a capacitor in parallel, thus forming a resonant circuit. At the center frequency of resonance, the maximum magnetic field is achieved at both the transmitting coil and the receiving coil. The inductor carries a resistance due to wiring. Thus controlling the characteristics of this RLC circuitry is the key to achieving higher bandwidth.

The basic MI principle used in [58] had limited application to low-power communication due to the small communication range, although it was higher than the range achieved with RF propagation underground with the same transmit power. However, this drastically changed with the introduction of the *MI waveguide* concept [61] in 2002. Using the regular transformer analogy, the basic idea is to create a chain of transformers in series: between the initial transmitting and receiving coils, guiding structures are used as MI relays, as illustrated in Fig. 8b. Such structures have essentially the same L and C elements of the MI-transmitter and receiver devices, excluding the active electronics interfaces. It is worth noting that MI waveguides are passive components and require no power source, a desirable feature for WUSNs.

In 2003, an empirical study [62] validated the physical model presented in [60, 61]. In 2007, the MI technology was highlighted as a potential option for WUSNs [63]. In the same year, MI was also considered for communication between implantable devices in the human body [64].

In 2009, the first communication models for UG2UG links in WUSNs were proposed in [65]. Both the *ordinary* (basic) MI channel and MI with waveguides were considered. Simulated results at 300 and 900 MHz were presented and it was reported that ordinary MI system had similar performance as EM-based solution. However, the MI technique is unaffected by variations in the permittivity of the medium

(i.e., soil). Moreover, with the use of the waveguide technique the path loss was greatly reduced. In 2010, the same authors presented more detailed versions of these MI models and simulations for 10 MHz were presented [66]. Different design parameters for the MI coil were considered and an inter-node distance of 250 m was achieved with a relatively small coil radius (0.15 m), 0.01 ohms/m wire for the coil, and MI relays (waveguides) placed every 5 m (49 relays). With configuration the bandwidth was limited to 1KHz. Changing the spacing of relays to 4 m (61 relays) increased the bandwidth to 2KHz.

The MI waveguide technique can allow more than one topology in a network. For instance, a relay coil can be used by more than one pair of MI transmitter/receiver. With this observation deployment algorithms were proposed in [67] along with simulated results. However, the implementation aspects of relay coils remain an open research topic.

In 2010, an additional MI communication model using waveguides was proposed [68]. Three excitation methods were investigated. Differently from underground communication models for EM wave propagation, the MI models are relatively simple because the environment usually does not play a significant role in this case. However, if there are underground metallic objects, a modified MI model or architecture becomes necessary.

In 2010 and 2011 some MI models were proposed for specific scenarios, e.g., underground network formed around pipelines [2, 69, 70]. In [2], a WUSN architecture (MI-based) for underground pipeline monitoring was proposed. Relay coils were suggested to extend the communication range *only* if the pipeline was made of non-metallic material. However, it was highlighted that for metallic pipelines, no (or very few) relay coils are necessary because the pipeline itself would provide the magnetic core for the MI waveguide. In [69], a similar WUSN scenario with steel pipes for controlling pumps in a heating system was discussed. The pipelines were metallic and a theoretical model was developed with simulated results. A low frequency of 3KHz was used in the design/simulations. The maximum coverage varies from 20 to 40 m for a total path loss of 100 dB at this frequency. More recently in [70], the authors presented the optimal design for transmitter and receiver coil antennas for the system proposed in [69] at 3 KHz, along with an interesting discussion on the advantages and disadvantages of using the metallic pipe as core for the coils, one design guideline of which being that the transmitting coil should have the smallest possible radius and the converse for the receiving coil. The authors ultimately decided to air-filled core coils.

In 2012, rescue system for miners (point-to-point communication) based on MI is presented in [47]. Communication range of up to 30 m is achieved with a relative low-power device and triaxial antenna loops.

It is important to highlight that the physical dimensions of the antenna in [47] is around 30 cm^3. In fact, based on the design of other MI antennas at the related works listed in this chapter, we can expect relatively higher volumetric dimensions of a MI antenna compared with the antennas used in typical WSNs and also in EM-based WUSN nodes. Nonetheless, despite this potential constraint and also reduced bandwidth, the MI technique is a very promising one for WUSNs. Ordinary MI

technology is already commercially available for the military and the mining industry, in the form of point-to-point communication. The adaptation of such solutions to a network, including the MI relay devices, is the main challenge in using MI in WUSNs. For instance, MI devices behave similarly to directional antennas, leading to network coverage issues that need to be investigated.

3.2 Comparing EM and MI-Based Solutions for WUSNs

In this section, EM and MI techniques are compared in relation to their application to the physical layer of UG2UG communication in WUSNs. This discussion is not meant to be conclusive as technologies continue to evolve. The enhancements to both techniques, LW and MI waveguide, respectively, have not been empirically validated in large-scale networks. For certain WUSN scenarios, one or the other solution may be a better choice. The following are some of the important aspects that highlight the pros and cons of EM and MI solutions:

Communication with aboveground nodes: While an EM transceiver can be used for both UG2AG and AG2UG communication with relative long distances, the communication range of MI devices for the over-the-air path is significantly impacted. Therefore, in many UG2AG/AG2UG scenarios presented in this chapter, the EM-based solution is the answer.

Communication range: An ordinary MI system is typically superior compared to the basic EM solution in terms of communication range. This is even more prominent when we also consider the fact that EM wave propagation is highly impacted by soil parameters, in particular soil moisture. On the other hand, the LW effect is expected to improve EM systems, while implementation guidelines on MI systems have yet to appear. Therefore, it is at present unclear if MI systems will maintain superiority in range. For instance, a critical aspect in installing relay coils is the need for proper alignment among the coils, which counts against MI systems.

Network topology: The directionality of MI antennas is evident in Fig. 8b. Therefore, chain-like deployments are ideal to be realized by MI-based solutions. On the other hand, if disk-like coverage is an important consideration, MI systems can be more complex than desired. EM solutions are more flexible with respect to this requirement, depending on the type of antenna employed.

Bandwidth: EM solutions typically have a small bandwidth and resulting data rate (e.g., 250 Kbps). Such data rate is usually enough for the majority of WUSN applications. On the other hand, the MI bandwidth is very small with an expected data rate of around 1 Kbps. A significant number of WUSN applications will not operate properly under this constraint. This is one of the main limitations of MI solutions applied to WUSNs.

Energy: The comparisons between EM and MI in this chapter have been made assuming the same low-power regime (e.g., between 0 and 10 dBm). However, MI solutions are potentially more energy efficient. This is particularly true when the soil moisture level increases. In such cases, EM transceivers are forced to temporarily use

a higher transmission power. Alternatively, communication can be suspended while VWC is high. MI solutions are not significantly impacted by the environmental conditions underground. Moreover, the relay coils used in the MI waveguide are passive elements.

Deployment costs: Judging from existing MI literature, the dimensions of MI devices are significantly larger compared to EM devices. Therefore, in cases where a hole must be dug to install MI devices, the associated costs are expected to be higher. This is especially true if shallower deployments are not an option or are less desirable, as the effects of installing MI devices close to soil surface have not been investigated so far.

Transceivers: Commodity RF transceivers are used for EM-based WUSNs and usually the antenna is the component that requires additional customization. However, low-power MI transceivers are not commercially available and such design must be included in the current MI implementation.

Air-filled underground scenarios: Many of the scenarios where MI-based solutions are being proposed are building basements and similar environment. In these cases, the LW effect cannot be used as a factor to boost the communication range for EM-based solutions. Consequently, MI devices are potentially the proper choice.

Enhancements to EM-based solutions include: (a) the adoption of shallower depths when possible, (b) antenna design, and (c) the adoption of environment-aware communication protocols [5, 6, 8, 11, 12, 71]. By contrast, enhancements to MI-based solutions are unclear due to a lack of real-world implementation, especially involving MI waveguides. Thus the proper technology choice often depends on the state-of-the-art in EM and MI technologies at the time of the WUSN project. Nevertheless, there is no doubt that for some scenarios one of the two is a better option. For instance, when the aboveground communication is a regular requirement, EM is the proper choice. In confined large areas such as basements of buildings, MI technology can be a better option. In some other scenarios, a single technology may not be the answer and a hybrid solution may be better. The following case exemplifies why a hybrid solution is necessary in some cases.

Assume that soil sensors must be installed at 10, 70 and 90 cm depths on a farm. The sensors take soil measurements every 30 min. The sensors and the communication module must be protected against physical damage (due to the plowing machinery and similar activities). The proposed architecture mentioned in [12] is shown in Fig. 9. It is formed by a module close to the surface (topsoil), a module with a higher burial depth (subsoil), and aboveground devices. The subsoil device has a permanent installation and has an expected lifetime of 3 or more years. The topsoil devices have temporary installation, that is, they are manually installed just after the seeding phase. Also, these devices are removed right before crop harvesting.

In this case, clearly the use of MI devices alone is not enough due to the existence of aboveground communication. Similarly, EM technology alone is problematic due to the energy requirements and the high soil path losses involved. Therefore, a hybrid solution seems to be the proper design. The topsoil device can be an EM device installed very close to soil surface in order to extend the communication range. The energy solution for this device is not critical because the batteries can be replaced

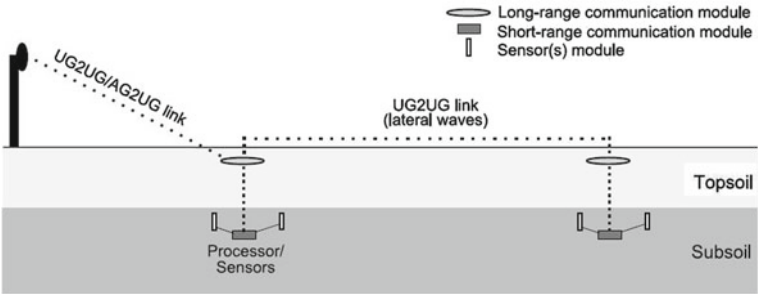


Fig. 9 Example of a scenario where a hybrid solution (EM and MI) is potentially the best architecture [12]

after every crop season. EM, however, is not a good solution to the subsoil module due to its deep installation. Even its communication with the topsoil module is a challenge when these 3 factors are considered simultaneously: low-power (constrained transmit power level), high soil moisture (irrigated area), and inter-node distance (e.g., >50 cm). Note that LWs is not an option for the UG2UG communication in this scenario due to the geometry involving the topsoil and subsoil devices. Thus the subsoil module should use MI technology instead to communicate with the topsoil device, which will have 2 transceivers (EM and MI). Recall that the soil moisture level will not significantly impact the MI path loss. Also, the inter-node distance is sufficiently small to guarantee very low power consumption while maintaining good communication. Finally, this application does not require high data rate and the small bandwidth of the MI communication will not be a problem.

As we can see from this example, both technologies (EM and MI) are being exploited considering their strengths and limitations. Therefore, before deciding on a single communication technology in the WUSN design, it is also helpful to evaluate if a hybrid solution exists.

This section is concluded with an overview of the wireless underground communication range achieved in empirical work, as shown in Table 3.

4 Research Directions and Conclusions

In this chapter, the challenges related to the underground deployment of a WSN (a WUSN) are investigated. Two technologies for the physical layer are considered: traditional wave propagation (EM) and the novel technique based on magnetic induction (MI). Both technologies have pros and cons. Based on simulated results, MI is clearly superior, as its communication performance is not significantly affected by environmental conditions including soil moisture variation. Also, in terms of communication range, MI is expected to have better performance, especially when MI waveguide is adopted. On the other hand, the current technology for the MI technique presents two main drawbacks: (a) it is associated with very small data rates,

Table 3 Communication range in wireless underground communication

UG→UG (EM)	UG→UG (MI)	UG→AG	AG→UG	Depth	Height (m)	Freq.	Range (m)	Ref.	Comments
✓				30 cm	-	27 MHz	20	[38]	17 % frame error rate for this distance
✓				40 cm	-	433 MHz	0.9	[8]	Clay soil, Mica2, 10 dBm
✓				1.4 m	-	2.4 GHz	1	[42]	Check the details of the experiment
	✓			?	-	2 KHz	35	[47]	Transmit power lever not informed
		✓		6 cm	1 m	2.4 GHz	7	[37]	MicaZ, 0 dBm
		✓		40 cm	12 m	868 MHz	30	[5]	10 dBm, special antenna scheme
		✓		25 cm	12m	868 MHz	150	[5]	1 Idem above
		✓	✓	10 cm	?	868 MHz	62	[51]	10 dBm
		✓		9 cm	?	2.4 GHz	∼100	[53]	19 dBm
		✓		10 cm	?	?	180	[48]	Nominal range (commercial product)
		✓		10 cm	?	900 MHz	152	[49]	Nominal range (commercial product)
			✓	35 cm	2.5	433 MHz	13	[10]	Clay soil, Mica2, 10 dBm, special antenna scheme
			✓	15 cm	2.5 m	433 MHz	38	[10]	Idem above
		✓		35 cm	2.5 m	433 MHz	22	[10]	Idem above
		✓		15 cm	2.5 m	433 MHz	30	[10]	Idem above

(b) the MI communication with aboveground devices can be severely impacted. The motivation for the latter statement is due to the fact that MI has a theoretical $1/r^3$ signal attenuation rate. In addition, the noise level of the air-channel is potentially higher for MI devices that operate at low frequencies, i.e., 50–20 KHz, compared to typical WSN radio channels. However, UG2AG and AG2UG links based on MI is still a research topic to be better investigated.

The lack of real-world implementation of large WUSNs based on MI contributes to the difficulty in some of our discussions and comparisons. Moreover, commodity MI devices are not yet available. On the other hand, EM solutions to WUSNs can be enhanced in a number of ways. The main technique is the combination of shallower burial depths and an antenna design tailored to the use of lateral waves (LW). However, similar to MI, empirical work exploiting such techniques is currently lacking. Based on the WUSN literature, we can expect to see more and more hybrid solutions mixing EM and MI technologies.

Once the physical layer of WUSNs achieves a high degree of acceptance and standardization, the next step will be development of network protocols tailored to WUSNs. This task is also a challenge because the required energy efficiency for

Table 4 Direction for future researches in WUSNs

Layer	Tech.	Open research challenge
Physical	EM	Antenna design specific for buried nodes (antenna factor)[5, 10–12, 23, 71, 72]
Physical	EM	Implementation of solutions based on LW effect [12–14, 45]
Physical	EM/MI	Modulation scheme [71]
Physical	EM	Soil channel-aware control error [10–12, 71, 72]
Physical	MI	Practical deployment strategies for MI waveguides [47, 69, 70]
Physical	EM/MI	Comparison between EM and MI techniques for UG2AG and AG2UG links
Physical	EM/MI	Hybrid EM/MI solutions
Data Link	EM	Soil channel-aware adaptive FEC schemes [71, 72]
Data Link	EM	Comparison between TDMA and contention-based MAC schemes [57, 72, 71]
Data Link	EM/MI	Optimal packet size [71, 72]
Network	EM	Soil channel-aware and event-aware adaptive routing protocols [71]
Network	EM/MI	Opportunistic routing protocols based on the use of aboveground nodes [12, 71]
Transport	EM	Transport reliability and QoS metrics specific for WUSNs [71]
Transport	EM	Congestion control for WUSNs [71]
Cross-layer	EM	Utilization of soil moisture (VWC) data for channel prediction [8, 12, 71]
Cross-layer	EM	Hybrid MAC/routing protocols [57, 71]
Cross-layer	EM	Hybrid Link/transport protocols [57, 71]

WUSNs is very critical. Besides hardware improvements, the design of software modules must consider the typical low data rate of WUSN applications, and a new protocol suite for WUSNs is required.

In a research work [71], the authors point out future directions for researches in WUSNs. At Table 4, we summarize these directions and include additional aspects considering both EM and MI systems. Observe that there are many issues related to the EM systems that were not listed for MI-based solutions, highlighting the fact that the MI technique is still in its first development stages.

References

1. I.F. Akyildiz, E.P. Stuntebeck, Wireless underground sensor networks: research challenges. Ad Hoc Netw. J. (Elsevier) **4**, 669–686 (2006)
2. Sun Z, Akyildiz I F (2009) Underground Wireless Communication using Magnetic Induction. In Proc. IEEE ICC 2009, Dresden, Germany.
3. M.C. Vuran, A.R. Silva, *Communication Through Soil in Wireless Underground Sensor Networks—Theory and Practice. Where Theory Meets Practice* (Springer, Berlin, 2009)
4. M.J. Tiusanen, Attenuation of a Soil Scout radio signal. Biosyst. Eng. **90**(2), 127–133 (2005)
5. M.J. Tiusanen, Wideband antenna for underground Soil Scout transmission. IEEE Antennas Wirel. Propag. Lett. **5**(1), 517–519 (2006)
6. M.J. Tiusanen, Wireless Soil Scout prototype radio signal reception compared to the attenuation model. Precis. Agric. **10**(5), 372–381 (2008)
7. H.R. Bogena et al., Hybrid wireless underground sensor networks: quantification of signal attenuation in soil. Vadose Zone J. **8**(3), 755–761 (2009)
8. A.R. Silva, M.C. Vuran, Empirical evaluation of wireless underground-to-underground communication in wireless underground sensor networks, in *Proceedings of IEEE DCOSS '09*, Marina Del Rey, CA (2009)
9. A.R. Silva, M.C. Vuran, Development of a Testbed for Wireless Underground Sensor Networks. EURASIP J. Wirel. Commun. Netw. **2010**, 1–15 (2010)
10. A.R. Silva, M.C. Vuran, Communication with aboveground devices in wireless underground sensor networks: an empirical study, in *Proceedings of IEEE ICC '10*, Cape Town, South Africa (2010)
11. A.R. Silva, M.C. Vuran, $(CPS)^2$: integration of center pivot systems with wireless underground sensor networks for autonomous precision agriculture, in *ACM/IEEE International Conference on Cyber-physical Systems (ICCPS '10)*, Stockholm, Sweden (2010)
12. A.R. Silva, Channel characterization for wireless underground sensor networks. Master's thesis, University of Nebraska at Lincoln (2010)
13. X. Dong, M.C. Vuran, A channel model for wireless underground sensor networks using lateral waves, in *Proceedings of IEEE Globecom 2011*, Houston, TX
14. Z. Sun et al., Dynamic connectivity in wireless underground sensor networks. IEEE Trans. Wirel. Commun. **10**(12), 4334–4344 (2011)
15. Z. Sun et al., BorderSense: border patrol through advanced wireless sensor networks. Ad Hoc Netw. J. (Elsevier) **9**(3), 468–477 (2011)
16. H.D. Foth, *Fundamentals of Soil Science*, 8th edn. (Wiley, New York, 1990)
17. L.K. Bandyopadhyay et al., *Wireless Communication in Underground Mines: RFID-based Sensor Networking* (Springer, New York, 2010)
18. J. Behari, *Microwave Dielectric Behavior of Wet Soils* (Springer, New Delhi, 2005)
19. A. Chukhlantsev, *Microwave Radiometry of Vegetation Canopies* (Springer, Netherlands, 2006)
20. H.D. Foth, *Fundamentals of Soil Science*, 8th edn. (Wiley, Canada, 1990)

21. W.H. Gardner, Water content, ed. by A. Klute, *Methods of Soil Analysis. Part 1*, 2nd edn. (American Society of Agronomy, Soil Science Society of America, Madison, 1986)
22. T.S. Rappaport, *Wireless Communications: Principles and Practice*, 1st edn. (Prentice Hall PTR, New Jersey, 1996)
23. R. King et al., *Antennas in Matter—Fundamentals, Theory, and Applications* (MIT Press, Massachusetts, 1981)
24. N. Peplinski et al., Dielectric properties of soils in the 0.3-1.3-Ghz range. IEEE Trans. Geosci. Remote Sens. **33**(3), 803–807 (1995)
25. L. Li et al., Characteristics of underground channel for wireless underground sensor networks, in *Proceedings of Med-Hoc-Net 07*, Corfu, Greece (2007)
26. I.F. Akyildiz et al., Signal propagation techniques for wireless underground communication networks. Phys. Commun. J. (Elsevier) **2**(3), 167–183 (2009)
27. A. Sommerfeld, Uber die ausbreitung der wellen in der drahtlosen telegraphie (About the propagation of waves in wireless telegraphy). Ann. Physik **28**, 665–737 (1909)
28. C.T. Tai, Radiation of a Hertzian dipole immersed in a dissipative medium. Cruft Laboratory Technical Report 21, Harvard University (1947)
29. C.T. Tai, R. Collin, Radiation of a hertzian dipole immersed in a dissipative medium. IEEE Trans. Antennas Propag. **48**(10), 1501–1506 (2000)
30. A. Banos, *Dipole Radiation in the Presence of a Conducting Half-Space* (Pergamon Press, Oxford, 1966)
31. L.M. Brekhovskikh, *Waves in Layered Media*, 2nd edn. (Academic Press, New York, 1980)
32. D.J. Daniels, Surface-penetrating radar. Commun. Eng. J. **8**(4), 165–182 (1996)
33. T.R.H. Holmes, Measuring surface soil parameters using passive microwave remote sensing. The Elbara field campaign 2003. Master's thesis, Vrije Universiteit Amsterdam (2003)
34. T.W. Miller et al., Effects of soil physical properties on GPR for landmine detection, in *Proceedings of the Fifth International Symposium on Technology and the Mine Problem*, Monterey, CA (2002)
35. T.P. Weldon, A.Y. Rathore, Wave propagation model and simulations for landmine detection. Technical report, University of N. Carolina at Charlotte (1999)
36. E. Odei-Lartey, K. Hartmann, Wireless ad hoc and sensor network underground with sensor data in real-time, in *Proceedings of ICSNC 2011*, Barcelona, Spain (2011)
37. E. Stuntebeck et al., Underground wireless sensor networks using commodity terrestrial motes, in *Poster Presentation at IEEE SECON 2006*, Reston, USA (2006)
38. J. Huang et al., Development of a wireless soil sensor network, in *2008 ASABE Annual International Meeting*, Providence, Rhode Island (2008)
39. A. Valera et al., Underground wireless communications for monitoring of drag anchor embedment parameters: a feasibility study, in *Proceedings of AINA '10 Advanced Information Networking and Applications*, pp. 713–720
40. L. Montrasio, G. Ferrari, A distributed wireless soil displacement measurement system for active monitoring of the excavation front of a gallery. Open Electr. Electron. Eng. J. **5**, 1–8 (2011)
41. W. Quinn et al., Design and performance analysis of an embedded wireless sensor for monitoring concrete curing and structural health. J. Civil Struct. Health Monit. **1**, 47–59 (2011)
42. S. Yoon et al., A radio propagation model for wireless underground sensor networks, in *Proceedings of IEEE Globecom* (2011)
43. H. Xiaoya et al., Channel modeling for wireless underground sensor networks, in *Proceedings of Computer Software and Applications Conference Workshops (COMPSACW), 2011 IEEE 35th Annual*, pp. 249–254 (2011)
44. D.M. Schwartz, Antenna and radio wave propagation characteristics at VHF near and in the ground. Master's thesis, University of Texas (1963)
45. R. King et al., Lateral electromagnetic waves: theory and applications to communications, geophysical exploration, and remote sensing (Springer, Heidelberg, 1992)
46. Ultra Electronics Inc. Magneto Inductive Remote Activation Munition System (MIRAMS). http://www.ultra-ms.com/capabilities/through-the-earth-communication/mirams.html. Accessed 23 April 2012

47. A. Markham, N. Trigoni, Magneto-inductive networked rescue system (MINERS): taking sensor networks underground, in *Proceedings of IEEE IPSN*, Beijing, China, pp. 317–328 (2012)
48. ProHome Soil-sensor-system wireless. http://www.ugmo.com/products/prohome. Accessed 23 April 2012
49. TurfGuard System. http://www.toro.com/irrigation/golf/turfguard/micro/index.html. Accessed 23 April 2012
50. A. Ahmed et al., Experiment measurements for packet reception rate in wireless underground sensor networks, in *Proceedings of ICOCI' 09*, Kuala Lumpur, Malaysia (2009)
51. C. Ritsema et al. A new wireless underground network system for continuous monitoring of soil water contents. Water Resour. Res. **45**, W00D36, 9 (2009)
52. M. Lin et al., Wireless sensor network: water distribution monitoring system, in*Proceedings of IEEE Radio and Wireless Symposium*, Orlando, Florida (2008)
53. H.R. Bogena et al., Potential of of wireless sensor networks for measuring soil water content variability. Vadose Zone J. **9**(4), 1002–1013 (2010)
54. A.R. Silva, M.C. Vuran, Empirical Evaluation of Wireless Underground-to-Aboveground Communication, in *Poster Session IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS '09)*, Marina Del Rey, CA (2009)
55. A. Adel, F. Norsheila, Probabilistic routing protocol for a hybrid wireless underground sensor networks. Wirel. Commun. Mobile Comput. (2011). doi:10.1002/wcm.1101
56. X. Wu, M. Liu, In-situ soil moisture sensing: measurement scheduling and estimation using compressive sensing, in *Proceedings of IPSN*, New York, NY, USA, pp. 1–12 (2012)
57. A. Silva, M. Liu, M. Moghaddam, Ripple-2: a non-collaborative; asynchronous; and open architecture for highly-scalable and low duty-cycle WSNs, in *Proceedings of ACM MiSeNet*, Istanbul, Turkey, pp. 39–44 (2012)
58. J.J. Sojdehei et al., Magneto-inductive (MI) communications, in *Proceedings of OCEANS, 2001 MTS/IEEE Conference and Exhibition*, vol. 1, pp. 513–519 (2001)
59. C. Bunszel, Magnetic induction: a low-power wireless alternative. RF Des. **24**(11), 78–80 (2001)
60. E. Shamonina et al., Magneto-inductive waves in one, two, and three dimensions. J. Appl. Phys. **92**, 6252–6261 (2002)
61. E. Shamonina et al., Magneto-inductive waveguide. Electron. Lett. **38**, 371–373 (2002)
62. M. Wiltshire et al., Dispersion characteristics of magneto-inductive waves: comparison between theory and experiment. Electron. Lett. **39**, 215–217 (2003)
63. N. Jack, K. Shenai (2007) Magnetic Induction IC for Wireless Communication in RF-Impenetrable Media, *Proceedings of IEEE Workshop on Microelectronic and Electron Devices*, Boise, ID
64. Sun M et al., How to pass information and deliver energy to a network of implantable devices within the human body, in *Proceedings of IEEE Engineering in Medicine and Biology Society*, Lyon, France, pp. 5286–5289 (2007)
65. J. Agbinya, M. Masihpour, Excitation methods for magneto inductive waveguide communication systems, in *Proceedings of Fifth International Conference on Broadband and Biomedical Communications*, Malaga, Spain, pp. 1–6 (2010)
66. Z. Sun et al., MISE-PIPE: magnetic induction-based wireless sensor networks for underground pipeline monitoring. Ad Hoc Netw. J. (Elsevier) **9**(3), 218–227 (2011)
67. S.A. Meybodi et al., Magneto-inductive underground communications in a district heating system, in *Proceedings of ICC' 11*, Kyoto, Japan, pp. 1–5 (2011)
68. S.A. Meybodi et al., Magneto-inductive communication among pumps in a district heating system, in *2010 9th International Symposium on Antennas Propagation and EM Theory (ISAPE)*, pp. 375–378 (2010)
69. Z. Sun, I.F. Akyildiz, Magnetic induction communications for wireless underground sensor networks. IEEE Trans. Antennas Propag. **58**(7), 2426–2435 (2010)
70. Z. Sun, I.F. Akyildiz, Deployment algorithms for wireless underground sensor networks using magnetic induction, in *Proceedings of IEEE GLOBECOM 2010*, Miami (2010)

71. I. Akyildiz, M. Vuran, *Wireless Sensor Networks, Series in Communications and Networking*, vol. 6. (Wiley, New York, 2010). ISBN 9780470036013
72. J. Tooker, M. Vuran, mobile data harvesting in wireless underground sensor networks, in *IEEE SECON*, Seoul, Korea, pp. 560–568 (2012)

Chapter 13
A Communication Framework for Networked Autonomous Underwater Vehicles

Baozhi Chen and Dario Pompili

Abstract Underwater acoustic communications consume a significant amount of energy due to the high transmission power (10–50 W) and long data packet transmission duration (0.1–1 s). Mobile Autonomous Underwater Vehicles (AUVs) can conserve energy by waiting for the 'best' network topology configuration, e.g., a *favorable alignment*, before starting to communicate. Due to the frequency-selective underwater acoustic ambient noise and high medium power absorption—which increases exponentially with distance—a shorter distance between AUVs translates into a lower transmission loss and a higher available bandwidth. By leveraging the predictability of AUV trajectories, a novel solution is proposed that optimizes communications by delaying packet transmissions in order to wait for a favorable network topology (thus trading end-to-end delay for energy and/or throughput). In addition, the proposed solution exploits the frequency-dependent radiation pattern of underwater acoustic transducers to reduce communication energy consumption. Our solution is implemented and evaluated through emulations, showing improved performance over some well-known geographic routing solutions and delay-tolerant networking solutions.

1 Introduction

UnderWater Acoustic Sensor Networks (UW-ASNs) [1] have been deployed to carry out collaborative monitoring tasks including oceanographic data collection, disaster prevention, and navigation. To enable advanced underwater explorations, Autonomous Underwater Vehicles (AUVs), equipped with underwater sensors, are used for information gathering. Underwater *gliders* are one type of battery-powered

B. Chen(✉) · D. Pompili
Department of Electrical and Computer Engineering, Rutgers University,
New Brunswick, NJ, USA
e-mail: baozhi_chen@cac.rutgers.edu

H. M. Ammari (ed.), *The Art of Wireless Sensor Networks*,
Signals and Communication Technology, DOI: 10.1007/978-3-642-40066-7_13,

energy-efficient AUVs that use hydraulic pumps to vary their volume in order to generate the buoyancy changes that power their forward gliding. These gliders are designed to rely on local intelligence with minimal onshore operator dependence. Acoustic communication technology is employed to transfer vital information (data and configuration) among gliders underwater and, ultimately, to a surface station where this information is gathered and analyzed.

Position information is of vital importance in mobile underwater sensor networks as the collected data has to be associated with appropriate location in order to be spatially reconstructed onshore. Even though AUVs can surface periodically (e.g., every few hours) to locate themselves using Global Positioning System (GPS)—which does not work underwater—over time inaccuracies in models for deriving position estimates, self-localization errors, and drifting due to ocean currents will lead to the increase of position uncertainty of underwater vehicle. Such uncertainty may degrade the quality of collected data and also the efficiency, reliability, and data rates of underwater inter-vehicle communications [39]. Besides the need to associate sensor data with 3D positions, position information can also be helpful for underwater communications. For example, underwater geographic routing protocols (e.g., [23, 25]) assume the positions of the nodes are known. AUVs involved in exploratory missions usually follow predicable trajectories, e.g., gliders follow *saw-tooth* trajectories, which can be used to predict position and, therefore, to improve communication.

By leveraging the predictability of the AUVs' trajectory, the energy consumption for communication can be minimized by delaying packet transmissions in order to wait for a *favorable network topology*, thus trading end-to-end (e2e) delay for energy and/or throughput.[1] For instance, Fig. 1 depicts a scenario where glider i waits for a certain time period Δt [s] to save transmission energy and to achieve higher throughput. Based on j's and d's trajectory, glider i predicts a 'better' topology with shorter links after Δt and postpones transmission in favor of lower transmission energy and higher data rate. This approach differs from that proposed for Delay Tolerant Networks (DTNs), where delaying transmission becomes necessary to overcome the temporary lack of network connectivity [11].

To estimate an AUV's position, in [9] we proposed a statistical approach to estimate a glider's trajectory. The estimates were used to minimize e2e energy consumption for networks where packets in the queue need to be forwarded right away (delay-sensitive traffic). In this work, we focus on delay-tolerant traffic and propose an optimization framework that uses acoustic directional transducers to reduce the computation and communication overhead for inter-vehicle data transmission. Moreover, we offer the distinction between two forms of position uncertainty depending on the network point of view, i.e., *internal* and *external uncertainty*, which refer to the position uncertainty associated with a particular entity/node (such as an AUV) as seen *by itself* or *by others*, respectively (see Sect. 5.1 for more details).

[1] Due to the peculiar 'V' shape of the underwater acoustic ambient noise and the high medium power absorption exponentially increasing with distance [35], a shorter distance between AUVs translates into a lower transmission loss and a higher available bandwidth.

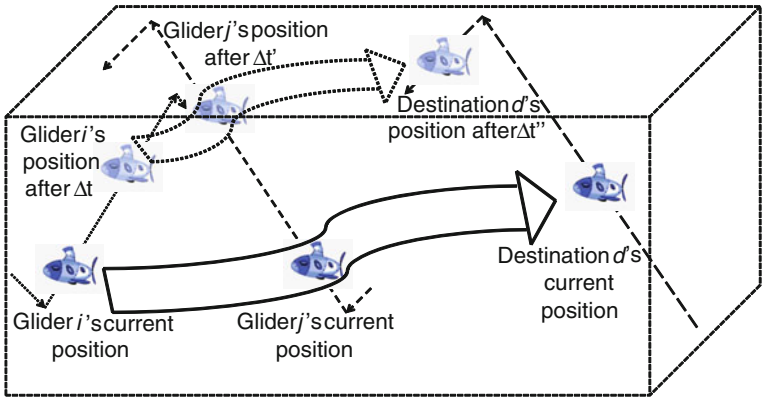


Fig. 1 Glider i delays its transmission by Δt waiting for a better topology so to improve e2e energy and/or throughput to destination d. *Wide arrows* represent the packet forwarding routes and *dashed/dotted simple arrows* represent glider trajectories

Based on the estimated external uncertainty, we propose QUO VADIS,[2] a **Q**oS-aware **u**nderwater **o**ptimization framework for inter-**v**ehicle communication using **a**coustic **di**rectional transducer**s**. QUO VADIS is a cross-layer optimization framework for delay-tolerant UW-ASNs that jointly considers the e2e delay requirements and constraints of underwater acoustic communication modems, including transducer directivity, power control, packet length, modulation, and coding schemes. Specifically, the proposed framework uses the external-uncertainty region estimates of the gliders and forwards delay-tolerant traffic with large maximum e2e delay, which includes *Class I (delay-tolerant, loss-tolerant)* traffic and *Class II (delay-tolerant, loss-sensitive)* traffic [25]. Moreover, our cross-layer communication framework exploits the frequency-dependent radiation pattern of underwater acoustic transducers. By decreasing the frequency band, transducers can change their "directivity" turning from being almost omnidirectional (with a gain of $\approx$0 dBi)—which is a desirable feature to support neighbor discovery and multicasting, geocasting, anycasting, and broadcasting—to directional (with gains up to 10 dBi)—which is useful for long-haul unicast transmissions.

The contributions of this work are as follows:

- We offer the distinction between two forms of position uncertainty (internal and external, depending on the view of the different nodes). A statistical approach is then proposed to estimate the position uncertainty and this estimated uncertainty is then used to improve network performance.
- We exploit the frequency-dependent directivity of the acoustic transducer that is originally used as omnidirectional transducer at one frequency to optimize network performance.

[2] "Quo vadis?" is a Latin phrase meaning "Where are you going?".

- We propose a distributed communication framework for delay-tolerant applications where AUVs can conserve energy by waiting for a 'good' network topology configuration, e.g., a *favorable alignment*, before starting to communicate.

The remainder of this chapter is organized as follows. We first introduce the basic knowledge on underwater acoustic sensor networks in Sect. 2 and review the related work in Sect. 3. Then we present the underwater communication model in Sect. 4 and propose our solution, QUO VADIS, in Sect. 5. In Sect. 6, performance evaluation and analysis are carried out, while conclusions are discussed in Sect. 7.

2 Basics of Underwater Acoustic Sensor Networks

UW-ASNs are applied in a broad range of applications, including environmental monitoring, undersea exploration, disaster prevention, assisted navigation and tactical surveillance.

Underwater networking is a rather unexplored area although underwater communications have been experimented since World War II, when, in 1945, an underwater telephone was developed in the United States to communicate with submarines [29]. Acoustic communications are the typical physical layer technology in underwater networks. In fact, radio waves propagate at long distances through conductive sea water only at extra low frequencies (30−300 Hz), which requires large antennae and high transmission power. For example, the Berkeley Mica2 Motes, the most popular experimental platform in the sensor networking community, have been reported to have a transmission range of 120 cm in underwater at 433 MHz by experiments performed at the Robotic Embedded Systems Laboratory (RESL) at the University of Southern California. Optical waves do not suffer from such high attenuation but are affected by scattering. Moreover, transmission of optical signals requires high precision in pointing the narrow laser beams. Thus, *acoustic waves* are generally used for underwater communications [34].

The traditional approach for ocean-bottom or ocean-column monitoring is to deploy underwater sensors that record data during the monitoring mission, and then recover the instruments [27]. This approach has the following disadvantages:

1. No real-time monitoring: The recorded data cannot be accessed until the instruments are recovered, which may happen several months after the beginning of the monitoring mission. This is critical especially in surveillance or in environmental monitoring applications such as seismic monitoring.

2. No online system reconfiguration: Interaction between onshore control systems and the monitoring instruments is not possible. This impedes any adaptive tuning of the instruments, nor is it possible to reconfigure the system after particular events occur.

3. No failure detection: If *failures* or *misconfigurations* occur, it may not be possible to detect them before the instruments are recovered. This can easily lead to the complete failure of a monitoring mission.

4. Limited storage capacity: The amount of data that can be recorded during the monitoring mission by every sensor is limited by the capacity of the onboard storage devices (memories, hard disks).

Therefore, there is a need to deploy underwater networks that will enable real-time monitoring of selected ocean areas, remote configuration and interaction with onshore human operators. This can be obtained by connecting underwater instruments by means of wireless links based on acoustic communication.

To communicate with each other acoustically, underwater sensor nodes need to use acoustic modems, which are able to convert electrical signals into sound waves and vice versa. As of today, many acoustic modems—such as those designed and manufactured by companies like LinkQuest, Teledyne Benthos, DSPComm are commercially available to provide communication capabilities in different underwater environments. These modems uses communication techniques such as Frequency-Shift Keying (FSK), Phase-Shift Keying (PSK), Direct Sequence Spread Spectrum (DSSS) and Orthogonal Frequency-Division Multiplexing (OFDM), offering data rates up to 38.4 kbps over different communication ranges, i.e., short range (up to about 500 m), medium range (up to about 4000 m), and long range (up to about 10000 m) in different underwater environments (shallow water or deep water) for different communication link setups (vertical or horizontal communication link).

These modems have been used in different underwater communication networks. However, they are generally big in size, which is not suitable for underwater vehicles such as the SLOCUM glider. Due to the size constraint, the popular choice for underwater gliders today is the Micro-Modem produced by Woods Hole Oceanography Institution (WHOI), as shown in Fig. 2. The WHOI Micro-Modem is currently the state-of-the-art modem used on the SLOCUM glider. It is compact in size (including the transducer), offering data rates from 80 to 5300 bps with communication range of up to a few kilometers. Such feature makes it an appropriate choice for AUVs like underwater gliders.

Many researchers are currently engaged in developing networking solutions for terrestrial wireless ad hoc and sensor networks. Although there exist many recently developed network protocols for wireless sensor networks, the unique characteris-

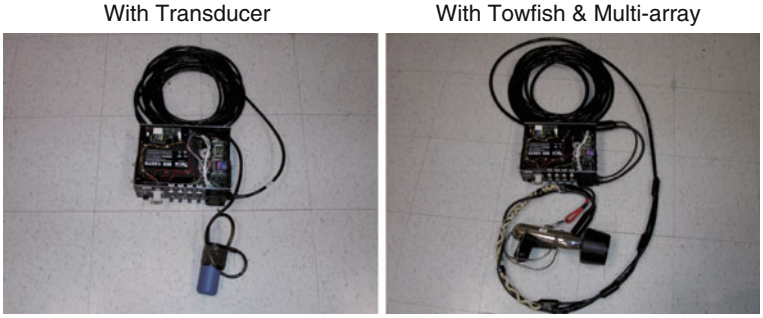


Fig. 2 WHOI micro-modem connected to different transducers

tics of the underwater acoustic communication channel, such as limited bandwidth capacity and variable delays [28], require very efficient and reliable new data communication protocols.

Major challenges in the design of underwater acoustic networks are as the following.

- The available bandwidth is severely limited;
- The underwater channel is severely impaired, especially due to multi-path and fading problems;
- Propagation delay in underwater is five orders of magnitude higher than in Radio Frequency (RF) terrestrial channels, and extremely variable;
- High bit error rates and temporary losses of connectivity (shadow zones) can be experienced, due to the extreme characteristics of the underwater channel;
- Battery power is limited and usually batteries can not be recharged, also because solar energy cannot be exploited;
- Underwater sensors are prone to failures because of fouling and corrosion.

Underwater acoustic communications are mainly influenced by *path loss*, *noise*, *multi-path*, *Doppler spread*, and *high and variable propagation delay*. All these factors determine the *temporal and spatial variability* of the acoustic channel, and make the available bandwidth of the *underwater acoustic channel* limited and dramatically dependent on both range and frequency. Long-range systems that operate over several tens of kilometers may have a bandwidth of only a few kHz, while a short-range system operating over several tens of meters may have more than a hundred kHz of bandwidth. In both cases these factors lead to low bit rate [7], in the order of tens of kbps for existing devices.

Here after we analyze the factors that influence acoustic communications in order to state the challenges posed by the underwater channels for underwater sensor networking. These include:

Path loss: *Attenuation* is mainly provoked by absorption due to conversion of acoustic energy into heat. The attenuation increases with distance and frequency. Figure 3 shows the acoustic attenuation with varying frequency and distance for a short range shallow water acoustic channel, according to the Urick's propagation model in [37] (see Sect. 4 for more details). The attenuation is also caused by scattering and reverberation (on rough ocean surface and bottom), refraction, and dispersion (due to the displacement of the reflection point caused by wind on the surface). Water depth plays a key role in determining the attenuation. *Geometric Spreading* refers to the spreading of sound energy as a result of the expansion of the wavefronts. It increases with the propagation distance and is independent of frequency. There are two common kinds of geometric spreading: *spherical* (omni-directional point source), which characterizes deep water communications, and *cylindrical* (horizontal radiation only), which characterizes shallow water communications.

Noise: *Man-made noise* is mainly caused by machinery noise (pumps, reduction gears, power plants), and shipping activity (hull fouling, animal life on hull, cavitation), especially in areas encumbered with heavy vessel traffic. *Ambient noise* is related to hydrodynamics (movement of water including tides, current, storms, wind,

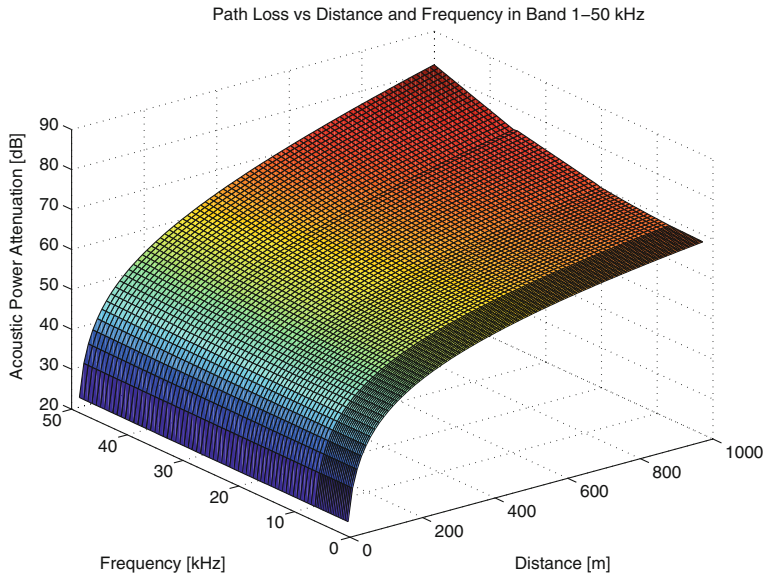


Fig. 3 Path loss of short range acoustic channel versus distance and frequency in band 1–50 kHz

and rain), and to seismic and biological phenomena. In [14], boat noise and snapping shrimps have been found to be the primary sources of noise in shallow water by means of measurement experiments on the ocean bottom.

Multi-path: Multi-path propagation may be responsible for severe degradation of the acoustic communication signal, since it generates Inter-Symbol Interference (ISI). The multi-path geometry depends on the link configuration. Vertical channels are characterized by little time dispersion, whereas horizontal channels may have extremely long multi-path spreads. The extent of the spreading is a strong function of depth and the distance between transmitter and receiver.

High delay and delay variance: The propagation speed in the acoustic channel is five orders of magnitude lower than in the radio channel. This large propagation delay (0.67 s/km) can reduce the throughput of the system considerably. The very high delay variance is even more harmful for efficient protocol design, as it prevents from accurately estimating the round trip time (RTT), which is the key parameter for many common communication protocols.

Doppler spread: The Doppler frequency spread can be significant in acoustic channels [34], causing a degradation in the performance of digital communications: transmissions at a high data rate cause many adjacent symbols to interfere at the receiver, requiring sophisticated signal processing to deal with the generated ISI. The Doppler spreading generates a simple frequency translation, which is relatively easy for a receiver to compensate for; and a continuous spreading of frequencies, which constitutes a non-shifted signal, which is more difficult to compensate for. If a channel has a Doppler spread with bandwidth B_{BW} and a signal has symbol duration T_{sym}, then there are approximately $B_{BW}T_{sym}$ uncorrelated samples of its

complex envelope. When $B_{BW}T_{sym}$ is much less than unity, the channel is said to be *underspread* and the effects of the Doppler fading can be ignored, while, if greater than unity, it is said to be *overspread* [17].

Most of the described factors are caused by the chemical-physical properties of the water medium such as temperature, salinity and density, and by their spatio-temporal variations. These variations, together with the wave guide nature of the channel, cause the acoustic channel to be *highly temporally and spatially variable*. In particular, the horizontal channel is by far more rapidly varying than the vertical channel, in both deep and shallow waters.

3 Related Work

We review the following areas: geographical routing solutions, terrestrial and underwater DTN solutions, solutions using directional transducers and underwater cross-layer optimization solutions, which are related to our work.

Geographic routing protocols rely on geographic position information for message forwarding, which requires that each node can determine its own location and that the source is aware of the location of the destination. In this way the message can be routed to the destination without knowledge of the network topology or *a priori* route discovery. Geographic routing protocols offer a number of advantages over conventional ad hoc routing protocols. Geographic routing does not require maintenance of routing tables or route construction prior to or during the forwarding process. Packet forwarding also allows a packet to adapt to topology change by selecting the next best hop based on the geographic location. It is also scalable as it does not rely on information that depends on the network size. Here we review some well-known geographic routing schemes that are proposed for terrestrial wireless networks as research on underwater geographic routing is still very limited. Many geographical routing schemes, including some well-known ones such as Most Forward within Radius (MFR) scheme [36], Greedy Routing Scheme (GRS) [12] and Compass Routing Method (CRM) [18], have been proposed for terrestrial wireless networks. In MFR, the message is forwarded to the neighbor that is closest to the destination, while in GRS a node selects the neighbor whose projection on the segment from the source to destination is closest to the destination (i.e., the node with maximum advance to the destination). In the Compass Routing Method (CRM) [18], a message is forwarded to a neighbor whose direction from the transmitter is the closest to the direction to the destination. In [22], a scheme called Partial Topology Knowledge Forwarding (PTKF) is introduced, and is shown to outperform other existing schemes in typical application scenarios. Based on the estimate using local neighborhood information, PTKF forwards packet to the neighbor that has the minimal e2e routing energy consumption. These solutions are proposed for terrestrial wireless networks. In UW-ASNs, they may not work well since propagation of acoustic signals is quite different from that of radio signals. Moreover, localization underwater is generally more difficult than in the terrestrial environment.

Delay tolerant networks are networks that have intermittent connectivity between network nodes, such as networks operating in mobile or extreme terrestrial environments, or interplanetary networks in deep space. In other words, DTNs are characterized by the lack of connectivity, resulting in a lack of instantaneous end-to-end paths. For networks using conventional protocols, such intermittent connectivity causes loss of data, where packets that cannot be forwarded immediately are dropped. For example, in TCP/IP networks, temporary disconnections may cause the slower packet retransmission. If packet dropping is too severe, TCP eventually ends the session, causing the applications to fail. To address this problem, protocols are designed carefully to support such intermittent communications between nodes in DTNs. Using the store-and-forward approach, a packet is incrementally moved and stored across the network so that it will eventually reach its destination. In this way, the reliability of packet forwarding can be guaranteed in DTNs. A common goal in many DTN routing protocols is to maximize end-to-end reliability. A common technique used to achieve this goal is to replicate copies of the message in the hope that it will succeed in reaching its destination.

Solutions for DTNs have been proposed for communications within extreme and performance-challenged environments where continuous e2e connectivity does not hold most of the time [5, 11]. Many approaches such as Resource Allocation Protocol for Intentional DTN (RAPID) routing [2], Spray and Wait [32], and MaxProp [4], are solutions mainly for intermittently connected terrestrial networks. RAPID [2] translates the e2e routing metric requirements such as minimizing average delay, minimizing worst-case delay, and maximizing the number of packets delivered before a deadline into per-packet utilities. At a transfer opportunity, it replicates a packet that locally results in the highest increase in utility. Spray and Wait [32] "sprays" a number of copies per packet into the network, and then "waits" until one of these nodes meets the destination. In this way it balances the tradeoff between the energy consumption incurred by flooding-based routing schemes and the delay incurred by spraying only one copy per packet in one transmission. MaxProp [4] prioritizes both the schedule of packets transmissions and the schedule of packets to be dropped, based on the path likelihoods to peers estimated from historical data and complementary mechanisms including acknowledgments, a head-start for new packets, and lists of previous intermediaries. It is shown that MaxProp performs better than protocols that know the meeting schedule between peers. These terrestrial DTN solutions may not achieve the optimal performance underwater as the characteristics of underwater communications are not considered. Hence, in the rest of this section, we focus on related solutions for UW-ASNs.

Several DTN solutions for UW-ASNs have been proposed in [8, 15, 16, 21]. In [8], an energy-efficient protocol is proposed for delay-tolerant data-retrieval applications. Efficient erasure codes and Low Density Parity Check (LDPC) codes are also used to reduce Packet Error Rate (PER) in the underwater environment. In [15], an adaptive routing algorithm exploiting message redundancy and resource reallocation is proposed so that 'more important' packets can obtain more resources than other packets. Simulation results showed that this approach can provide differentiated packet delivery according to application requirements and can achieve a good

e2e performance trade-off among delivery ratio, average e2e delay, and energy consumption. A Prediction Assisted Single-copy Routing (PASR) scheme that can be instantiated for different mobility models is proposed in [16]. An effective greedy algorithm is adopted to capture the features of network mobility patterns and to provide guidance on how to use historical information. It is shown that the proposed scheme is energy efficient and cognizant of the underlying mobility patterns.

In [21], an approach called Delay-tolerant Data Dolphin (DDD) is proposed to exploit the mobility of a small number of capable collector nodes (namely dolphins) to harvest information sensed by low power sensor devices while saving sensor battery power. DDD performs only one-hop transmissions to avoid energy-costly multi-hop relaying. Simulation results showed that limited numbers of dolphins can achieve good data-collection requirements in most application scenarios. However, data collection may take a long time as the nodes need to wait until a dolphin moves into the communication ranges of these nodes.

Compared to the number of approaches using directional antennae for terrestrial wireless sensor networks, solutions using directional transducers for UW-ASNs are very limited due to the complexity of estimating position and direction of vehicles underwater. Moreover, these solutions generally assume the transducers are ideally directional, i.e., they assume the radiation energy of the transducer is focused on some angle range with no leaking of radiation energy outside this range. For example, such transducers are used for localization using directional beacons in [19] and for directional packet forwarding in [40]. These solutions also use only one frequency. In our work, rather than using the ideal transducer model, we consider the radiation patterns of existing real-world transducers at different frequencies in order to minimize energy consumption for communications.

Over these years, cross-layer optimization becomes a popular choice to improve the performance in wireless networks. By removing the strict constraints on the communication interfaces between layers that are defined in the standard Open Systems Interconnection (OSI) model, different layers can share more information and interact with each other in order to improve the network performance. For example, in the physical layer, a node can change its channel coding based on the packet error rate from the link layer. Cross-layer optimization has been shown to be an effective way to improve the network performance, especially in a harsh environment such as the underwater [33]. A cross-layer optimization solution for UW-ASNs has been proposed in [25], where the interaction between routing functions and underwater characteristics is exploited, resulting in improvement in e2e network performance in terms of energy and throughput. Another cross-layer approach that improves energy consumption performance by jointly considering routing, MAC, and physical layer functionalities is proposed in [23]. These solutions, however, do not consider uncertainty in the AUV positions and are implemented and tested only by software simulation platforms and are not designed for delay-tolerant applications. On the contrary, we propose a practical uncertainty-aware cross-layer solution that incorporates the functionalities of the WHOI Micro-Modem [13] to minimize energy consumption. Moreover, our solution is implemented on real hardware and tested in our emulator integrating WHOI underwater acoustic modems.

4 Network Model

In this section we introduce the network model that our solution is based on and state the related assumptions. Suppose the network is composed of a number of gliders, which are deployed in the ocean for long periods of time (weeks or months) to collect oceanographic data. For propulsion, they change their buoyancy using a pump and use lift on wings to convert vertical velocity into forward motion as they rise and fall through the ocean. They travel at a fairly constant horizontal speed, typically 0.25 m/s [1]. Gliders control their heading toward predefined waypoints using a magnetic compass.

Assume the gliders need to forward the data they sensed to a collecting glider. The slow-varying and mission-dependent (and, for such reasons, 'predictable') trajectory of a glider is used in our solution to estimate another glider's position using the position and velocity estimate of some time earlier. A glider estimates its own trajectory and position uncertainty using its own position estimates; the parameters of the estimated trajectory and internal-uncertainty region are sent to neighboring gliders. Using these parameters, these gliders can extrapolate the glider's current position and a confidence region accounting for possible deviation from the extrapolated course.

The Urick model is used to estimate the transmission loss $TL(l, f)$ [dB] as,

$$TL(l, f) = \kappa \cdot 10 \log_{10}(l) + \alpha(f) \cdot l, \tag{1}$$

where l [m] is the distance between the transmitter and receiver and f [Hz] is the carrier frequency. Spreading factor κ is taken to be 1.5 for practical spreading, and $\alpha(f)$ [dB/m] represents an absorption coefficient that increases with f [35].

The Urick model is a coarse approximation for underwater acoustic wave transmission loss. In reality, sound propagation speed varies with water temperature, salinity, and pressure, which causes wave paths to bend. Acoustic waves are also reflected from the surface and bottom. Such uneven propagation of waves results in *convergence (or shadow) zones*, which are characterized by lower (or higher) transmission loss than that predicted by the Urick model due to the uneven energy dispersion.

Due to these phenomena, the Urick model is not sufficient to describe the underwater channel for simulation purposes. The Bellhop model is based on ray/beam tracing, which can model these phenomena more accurately. This model can estimate the transmission loss by two-dimensional acoustic ray tracing for a given sound-speed depth profile or field, in ocean waveguides with flat or variable absorbing boundaries. Transmission loss is calculated by solving differential ray equations, and a numerical solution is provided by HLS Research [26]. An example plotted using the Bellhop model is shown in Fig. 4. Interesting enough, if node 1 sends a packet, node 4 has higher probability of receiving the packet than node 3 even though this node is closer. Because the Bellhop model requires more information about the environment than a glider will have, such as sound speed profile and depths of receivers and ocean

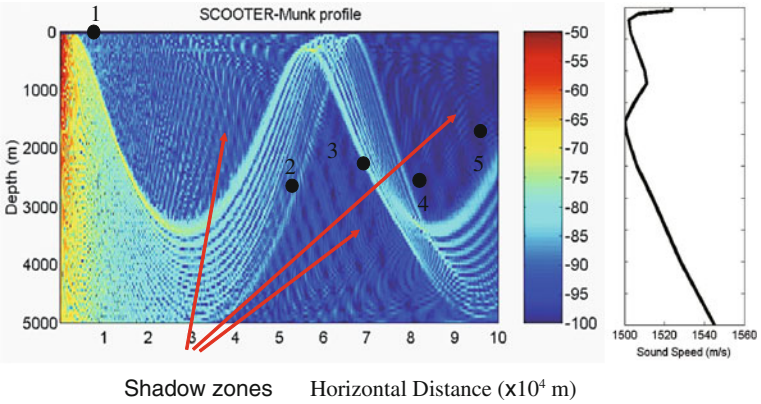


Fig. 4 Shadow zone scenario: the *left* subfigure represents the transmission loss of node 1 located at the origin, while the *right* subfigure depicts the sound speed profile used to derive the transmission loss (the y-axis is the depth, which has the same range used in the *left*; the *blue*, *yellow* and *red* areas denote large, medium and small path losses, respectively)

boundary, it is only used to simulate the acoustic environment for testing (relying on trace files with historic data). Hence, the proposed solution uses the Urick model in the cross-layer optimization (Sect. 5.2), which can be computed online on the glider.

We adopt the empirical ambient noise model presented in [35], where a 'V' structure of the power spectrum density (psd) is shown. The ambient noise power is obtained by integrating the empirical psd over the frequency band in use.[3]

5 Proposed Approach

Our proposed optimization is based on the estimation of the gliders' trajectories and their external-uncertainty regions. Therefore, in this section, we introduce the estimation of external-uncertainty regions for gliders first. We then present the cross-layer design of our proposed framework.

5.1 Internal and External Uncertainty

We first offer the distinction between two types of position uncertainty, followed by the discussion on the relationship between these two types of uncertainty. Then we present the statistical approach for external-uncertainty region estimation when

[3] Note that in underwater acoustics, power (or source level) is usually expressed using decibel (dB) scale, relative to the reference pressure level in underwater acoustics 1 μPa, i.e., the power induced by 1 μPa pressure. The conversion expression for the source level SL re μPa at the distance of 1 m of a compact source of P watts is $SL = 170.77 + 10 \log_{10} P$.

gliders are used as AUVs and ocean currents are unknown. Since the details have been presented in [10], we just summarize them here.

Internal uncertainty refers to the position uncertainty associated with a particular entity/node (such as an AUV) *as seen by itself*. Existing approaches such as those using Kalman Filter (KF) [3, 38] may not guarantee the optimality when the linearity assumption between variables does not hold. On the other hand, approaches using non-linear filters such as the extended or unscented KF attempt to minimize the mean squared errors in estimates by jointly considering the navigation location and the sensed states/features such as underwater terrain features, which are non-trivial, especially in an unstructured underwater environment.

External uncertainty, as introduced in this chapter, refers to the position uncertainty associated with a particular entity/node *as seen by others*. Let us denote the internal uncertainty, a 3D region associated with any node $j \in \mathcal{N}$($\mathcal{N}$ is the set of network nodes), as $\mathcal{U}_{jj}$, and the external uncertainties, 3D regions associated with j as seen by $i, k \in \mathcal{N}$, as $\mathcal{U}_{ij}$ and $\mathcal{U}_{kj}$, respectively ($i \neq j \neq k$). In general, $\mathcal{U}_{jj}, \mathcal{U}_{ij}$, and $\mathcal{U}_{kj}$ are different from each other; also, due to asymmetry, $\mathcal{U}_{ij}$ is in general different from $\mathcal{U}_{ji}$. External uncertainties may be derived from the broadcast/propagated internal-uncertainty estimates (e.g., using *one-hop or multi-hop neighbor discovery mechanisms*) and, hence, will be affected by e2e *network latency* and *information loss*.

The estimation of the external-uncertainty region $\mathcal{U}_{ij}$ of a generic node j at node i (with $i \neq j$) involves the participation of both i and j. Here we use the received $\mathcal{U}_{jj}$ as $\mathcal{U}_{ij}$ (a delayed version due to propagation delay, transmission delay and packet loss). Better estimation of $\mathcal{U}_{ij}$ involves estimation of the change of $\mathcal{U}_{jj}$ with time and is left as future work. We provide a solution for internal- and external-uncertainty estimation when (1) *gliders are used* (following a 'sawtooth' trajectory) and (2) *ocean currents are unknown*.

Internal-uncertainty estimation at j: Assume gliders estimate their own locations over time using *dead reckoning*. Given glider j's estimated coordinates, $P_n = (x_n, y_n, z_n)$ at sampling times t_n ($n = 1 \ldots N$), as shown in [10], its trajectory segment can be described as $P(t) = \bar{P} + \overrightarrow{\mathbf{v}}(t - \bar{t})$, where $\bar{P} = (\bar{x}, \bar{y}, \bar{z}) = \frac{1}{N}\sum_{n=1}^{N}(x_n, y_n, z_n)$ and $\overrightarrow{\mathbf{v}} = \frac{\|\overrightarrow{\widehat{P_1}\widehat{P_N}}\|}{\|(a^*,b^*,c^*)\|\cdot(t_N - t_1)} \cdot (a^*, b^*, c^*)$. Here, $[a^*, b^*, c^*]^T$ is the singular vector of $N \times 3$ matrix $\mathbf{A} = [[x_1 - \bar{x}, \ldots, x_N - \bar{x}]^T, [y_1 - \bar{y}, \ldots, y_N - \bar{y}]^T, [z_1 - \bar{z}, \ldots, z_N - \bar{z}]^T]$ corresponding to its largest absolute singular value, $\bar{t} = \frac{1}{N}\sum_{n=1}^{N} t_n$ is the average of the sampling times, and $\widehat{P_i}$ is the projection of point P_i on the line segment (Fig. 5a).

The internal-uncertainty region of j is estimated as a *cylindrical* region [10] $\mathcal{U}$ described by its radius R and its height $H_U - H_L$, where H_U and H_L—in general different—are the *signed distances* of the cylinder's top and bottom surface (i.e., the surface ahead and behind in the trajectory direction, respectively) to glider j's expected location on the trajectory. In [9] we demonstrate that:

1. H_L and H_U can be estimated as

$$\begin{cases} H_L = \overline{H} - \hat{t}_{\alpha,N-1} S^{(H)} \sqrt{1+1/N} \\ H_U = \overline{H} + \hat{t}_{\alpha,N-1} S^{(H)} \sqrt{1+1/N} \end{cases}, \tag{2}$$

where $\overline{H} = \sum_{n=1}^{N} H_n/N$ is the mean of these N samples, $S^{(H)} = [\frac{1}{N-1} \sum_{n=1}^{N} (H_n - \overline{H})^2]^{1/2}$ is the unbiased standard deviation, $1-\alpha$ is the confidence level, and $\hat{t}_{\alpha,N-1}$ is the $100(1-\alpha/2)$ % of *Student's t-distribution* [6] with $N-1$ degrees of freedom (here H_n is the nth sample calculated from P_n's [9]); and

2. R is estimated by

$$R = \frac{\sqrt{N-1} S^{(R)}}{\sqrt{\hat{\chi}_{\alpha,2(N-1)}}}, \tag{3}$$

where $S^{(R)} = [\frac{1}{N-1} \sum_{n=1}^{N} (R_n - \overline{R})^2]^{1/2}$, $\overline{R} = \frac{1}{N} \sum_{n=1}^{N} R_n$, and $\hat{\chi}_{\alpha,2(N-1)}$ is the $100(1-\alpha)$ % of χ*-distribution* with $2(N-1)$ degrees of freedom (here R_n is the nth sample calculated from P_n's [9]). As shown in Fig. 5b, j's internal-uncertainty region becomes smaller over time (from T_0 to T_2), i.e., as more position estimates are acquired.

External-uncertainty estimation at i: After receiving j's trajectory and internal-uncertainty region parameters ($\bar{P}$, $\bar{t}$, $\vec{\mathbf{v}}$, H_U, H_L, R), glider i can update the estimate of j's external-uncertainty region. Because AUVs involved in missions show predictable trajectories, information about the sawtooth segment can be used to derive the entire glider trajectory through extrapolation assuming symmetry between glider ascent and descent. Due to packet delays and losses in the network, j's external-uncertainty regions as seen by single- and multi-hop neighbors are *delayed versions* of j's own internal uncertainty (Fig. 5b). Hence, when using *multi-hop neighbor discovery schemes*, the internal uncertainty of a generic node j, $\mathcal{U}_{jj}$, provides a *lower bound* for all the external uncertainties associated with that node, $\mathcal{U}_{ij}$, $\forall i \in \mathcal{N}$. Hence we use the received $\mathcal{U}_{jj}$ as $\mathcal{U}_{ij}$ (a delayed version due to propagation delay, transmission delay and packet loss).

5.2 Cross-Layer Optimization for Delay-Tolerant Applications

With the external-uncertainty regions, a glider needs to select an appropriate neighbor to forward each packet to its final destination. Because the major part of available energy in battery-powered gliders should be devoted to propulsion [24], acoustic communications should not take a large portion of the available energy. Our proposed protocol minimizes the energy spent to send a message to its destination and considers the functionalities of a real acoustic modem for a practical solution. Specifically, we provide support and differentiated service to delay-tolerant applications with different QoS requirements, from loss sensitive to loss tolerant. Hence, we consider the following two classes of traffic:

Class I (delay-tolerant, loss-tolerant). It may include multimedia streams that, being intended for storage or subsequent offline processing, do not need to be deliv-

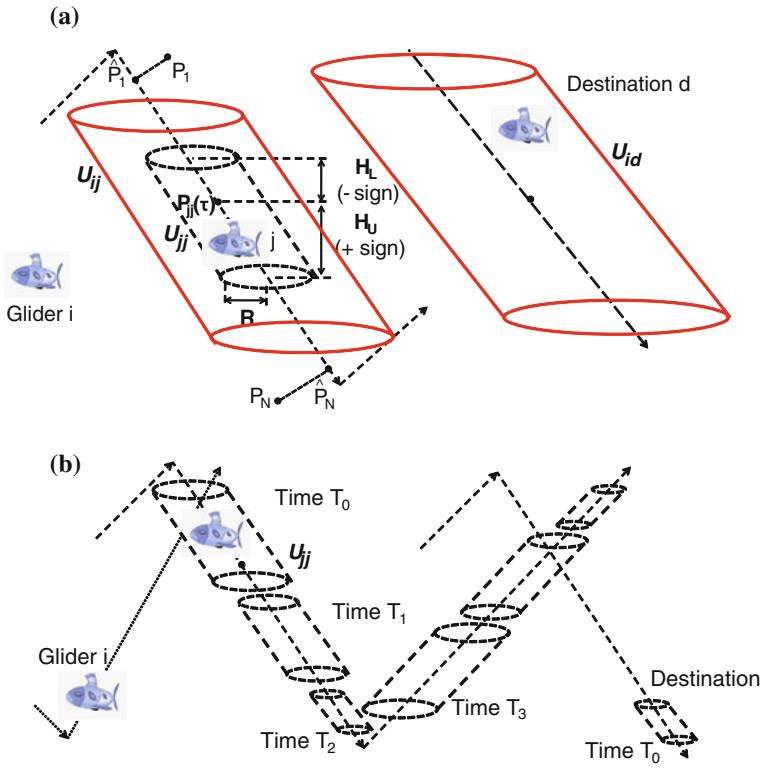


Fig. 5 External- and internal-uncertainty regions for gliders under the effect of unknown ocean currents. **a** Estimated internal-uncertainty region by j: a cylinder with circular *bottom* radius R and height $H_U - H_L$. **b** Change of internal-uncertainty region over time

ered within strict delay bounds. This class may also include scalar environmental data or non time-critical multimedia content such as snapshots. In this case, the loss of a packet is tolerable at the current hop, but its e2e PER should still be below a specified threshold.

Class II (delay-tolerant, loss-sensitive). It may include data from critical monitoring processes that require some form of offline post processing. In this case, a packet must be re-transmitted if it is not received correctly.

Our protocol employs only local information to make routing decisions, resulting in a scalable distributed solution (even though the destination information is required for routing, we can use the destination information learned from local neighbors to predict the position of the destination). It is a suboptimal solution instead of a global one since it relies on local information. The external-uncertainty regions obtained as described in Sect. 5.1 are used to select the neighbor with minimum packet routing energy consumption. Here, a framework using the WHOI Micro-Modem [13] is presented. This framework can be extended and generalized in such a way as to incorporate the constraints of other underwater communication modems.

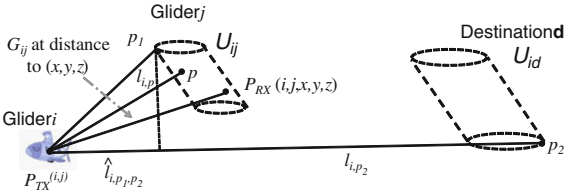


Fig. 6 Use of external-uncertainty region in the optimization framework

To be more specific, given the current time t_{now} [s] and a message m generated at time t_0 [s], glider i jointly optimizes the time Δt [s] to wait for the best topology configuration, a neighbor j^*, a frequency band f_{ij}, transmission power $P_{TX}^{(i,j)}(t)$ [W], packet type ξ, and number of frames[4] $N_F(\xi)$, so that the estimated energy $E_{id}(t)$ [J] to route m to destined glider d's region $\mathcal{U}_{id}$ is minimized and message m reaches it within $B_{\max}$ [s] (i.e., the maximum e2e delay from the source to the destination). We assume power control is possible in the range $[P_{min}, P_{max}]$ although transmission power is currently fixed for the WHOI Micro-Modem. We anticipate more advanced amplifier hardware will make this power optimization possible.

Here, $E_{id}(t)$ is estimated by the energy to transmit the packet to neighbor j in one transmission, the average number of transmissions $\hat{N}_{TX}^{(i,j)}(t)$ to send m to j, and the estimated number of hops $\hat{N}_{hop}^{(j,d)}(t)$ to reach region $\mathcal{U}_{id}$ via j. We need to estimate the transmission power and the number of hops to destination. The external-uncertainty region is used to estimate the number of hops $\hat{N}_{hop}^{(j,d)}(t)$ to d via neighbor j and the lower bound of the transmission power as follows (Fig. 6). Let $\hat{l}_{i,p_1,p_2}(t)$ [m] be the projected distance of line segment from i to position p_1 on the line from i to position p_2, and $l_{i,p}(t)$ be the distance from i to position p. $\hat{N}_{hop}^{(j,d)}(t)$ is estimated by the worst case of $l_{i,p}(t)/\hat{l}_{i,p_1,p_2}(t)$, i.e., Eq. (8). The lower bound for transmission power is estimated by the average transmission power so that the received power at every point in $\mathcal{U}_{ij}$ is above the specified threshold P_{TH}. The transmission power lower bound is the integral of the product of the transmission power to obtain P_{TH} at a point in $\mathcal{U}_{ij}$ and the probability density function (pdf) of j to be at this point.

To estimate the received power, it is necessary to estimate the transducer gains at the transmitter and receiver. To estimate the transmitter's gain $G_{TX}(\theta_{ij}, \phi_{ij}, f_{ij})$, i needs to compute the radiation angles—the horizontal angle $\theta_{ij} \in [-180°, 180°]$ and the vertical angle $\phi_{ij} \in [-90°, 90°]$ with respect to j. Assume the initial position of the transducer is as shown in the top left corner of Fig. 7, then i's normalized transducer direction vector is $\overrightarrow{\mathbf{n}_i} = (0, 0, -1)$ with the *horizontal plane* $z = z_0^{(i)}$ (defined as the plane perpendicular to $\overrightarrow{\mathbf{n}_i}$). While the glider is moving, its pitch, yaw, and roll angles are denoted by ε_i, ζ_i, and η_i, respectively. From geometry, the direction vector after rotation is $\overrightarrow{\mathbf{n}'_i} = \mathbf{Q}_x(\eta_i)\mathbf{Q}_y(\zeta_i)\mathbf{Q}_z(\varepsilon_i)\overrightarrow{\mathbf{n}_i}^T$, while the transducer's horizontal plane is $\mathbf{Q}_x(-\eta_i)\mathbf{Q}_y(-\zeta_i)\mathbf{Q}_z(-\varepsilon_i)[x, y, z]^T = z_0^{(i)}$, where $z_0^{(i)}$ is

[4] Each packet sent by WHOI Micro-Modem consists of a number of frames where the maximum number depends on ξ.

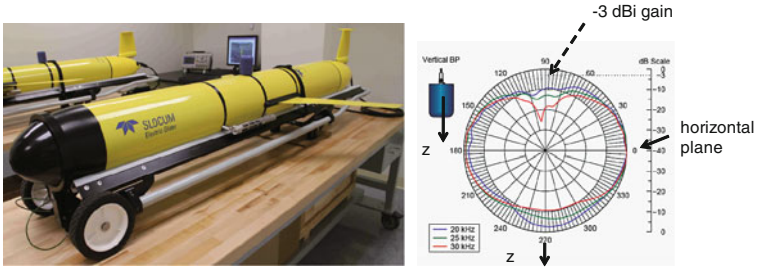


Fig. 7 Picture of our underwater glider and radiation pattern of the BT-25UR transduce

a constant, and $\mathbf{Q}_x(\eta_i)$, $\mathbf{Q}_y(\zeta_i)$ and $\mathbf{Q}_z(\varepsilon_i)$ are

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\eta_i & -\sin\eta_i \\ 0 & \sin\eta_i & \cos\eta_i \end{bmatrix}, \begin{bmatrix} \cos\zeta_i & 0 & -\sin\zeta_i \\ 0 & 1 & 0 \\ \sin\zeta_i & 0 & \cos\zeta_i \end{bmatrix}, \begin{bmatrix} \cos\varepsilon_i & -\sin\varepsilon_i & 0 \\ \sin\varepsilon_i & \cos\varepsilon_i & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

respectively.

With the position vector $\overrightarrow{P_iP_j}$ from i to j, we can derive $\cos\phi_{ij} = \frac{\widehat{\overrightarrow{P_iP_j}} \circ \overrightarrow{P_iP_j}}{\|\widehat{\overrightarrow{P_iP_j}}\| \cdot \|\overrightarrow{P_iP_j}\|}$ and $\cos\theta_{ij} = \frac{\widehat{\overrightarrow{P_iP_j}} \circ \vec{\mathbf{v}}_i}{\|\widehat{\overrightarrow{P_iP_j}}\| \cdot \|\vec{\mathbf{v}}_i\|}$, where $\widehat{\overrightarrow{P_iP_j}}$ is the projection of $\overrightarrow{P_iP_j}$ on the transducer's horizontal plane, $\circ$ is the inner product, and $\vec{\mathbf{v}}_i = \|\vec{\mathbf{v}}_i\| \cdot [\cos\varepsilon_i\cos\zeta_i, \cos\varepsilon_i\sin\zeta_i, \sin\varepsilon_i] = (a_i^*, b_i^*, c_i^*)$ is the velocity vector of glider i as estimated in Sect. 5.1. As $\vec{\mathbf{n}'}_i$ is perpendicular to the transducer's horizontal plane, we have $\sin\phi_{ij} = \cos(90 - \phi_{ij}) = \frac{\vec{\mathbf{n}'}_i \circ \overrightarrow{P_iP_j}}{\|\overrightarrow{P_iP_j}\|}$ and $\widehat{\overrightarrow{P_iP_j}} = \overrightarrow{P_iP_j} - (\overrightarrow{P_iP_j} \circ \vec{\mathbf{n}'}_i) \cdot \vec{\mathbf{n}'}_i$. The transducer's gain at receiver j, $G_{RX}(\theta_{ji}, \phi_{ji}, f_{ij})$, can be estimated in a similar way.

Let $L_m(\xi)$ be m's length in bits depending on packet type ξ and $B(\xi)$ be the corresponding bit rate. The energy to transmit the packet to neighbor j in one transmission can therefore be approximated by $P_{TX}^{(i,j)}(t) \cdot \frac{L_m(\xi)}{B(\xi)}$. Overall, the optimization problem can be formulated as

P(i, d, $\mathbf{t_{now}}$, $\Delta\mathbf{t_p}$): Cross-layer Optimization Problem

$$\begin{aligned}
&\textbf{Given:} P_{min}, P_{max}, \Xi, \Omega_\xi, G_{TX}, G_{RX}, \eta, B_{\max}, PER_{\max}^{e2e} \\
&\textbf{Computed:} \varepsilon_i, \zeta_i, \varepsilon_j, \zeta_j, \mathcal{U}_{ij}, \forall j \in \mathcal{N}_i \cup \{d\} \text{ (i.e., } R_j^{(i)}, H_L^{(i,j)}, H_H^{(i,j)}) \\
&\textbf{Find:} j^* \in \mathcal{N}_i, P_{TX}^{(i,j)*}(t) \in [P_{min}, P_{max}], \\
&\qquad \xi^* \in \Xi, N_F^*(\xi) \in \Omega_\xi, \Delta t^*, f_{ij}^* \in [f_L, f_U] \\
&\textbf{Minimize:} E_{id}(t) = P_{TX}^{(i,j)}(t) \cdot \frac{L_m(\xi)}{B(\xi)} \cdot \hat{N}_{TX}^{(i,j)}(t) \cdot \hat{N}_{hop}^{(j,d)}(t).
\end{aligned} \tag{4}$$

In $\mathbf{P(i, d, t_{now}}, \mathbf{\Delta t_p}, \mathcal{N}_i, \Xi$, and Ω_ξ denote the set of i's neighbors, the set of packet types, and the set of number of type ξ frames respectively. The objective function (4) estimates the energy required to send message m to the destination region $\mathcal{U}_{id}$. To solve this problem, we need to derive the relationship between these variables. Let $L_F(\xi)$ [bit] be the length of a frame of type ξ, L_H [bit] be the length of message m's header, $PER(SINR_{ij}(t), \xi)$ be the PER of type ξ at the Signal to Interference-plus-Noise Ratio $SINR_{ij}(t)$, $TL(l_{ij}(t), f_{ij})$ be the transmission loss for distance $l_{ij}(t)$ and carrier frequency f_{ij} [kHz]—which is calculated using Eq. (1)—$\mathcal{A}\backslash\{i\}$ be the set of active transmitters excluding i, and $P_{TX}^{(i,j)}(t)$ be the transmission power used by i to reach j, we have the following formulas,

(class-independent relationships)

$$t = t_{now} + \Delta t; \tag{5}$$

$$t_{TTL} = B_{\max} - (t_{now} - t_0); \tag{6}$$

$$L_m(\xi) = L_F(\xi) \cdot N_F(\xi) + L_H; \tag{7}$$

$$\hat{N}_{hop}^{(j,d)}(t) = \frac{\max_{p \in \mathcal{U}_{id}} l_{i,p}(t)}{\min_{p_1 \in \mathcal{U}_{ij}, p_2 \in \mathcal{U}_{id}} \hat{l}_{i,p_1,p_2}(t)}; \tag{8}$$

$$SINR_{ij}(t) = \frac{P_{TX}^{(i,j)}(t) \cdot 10^{G_{ij}(l_{ij}(t), f_{ij})/10}}{\sum_{k \in \mathcal{A}\backslash\{i\}} P_{TX}^{(k,j)}(t) \cdot 10^{G_{ij}(l_{kj}(t), f_{ij})/10} + N_0}; \tag{9}$$

$$G_{ij}(l_{ij}, f_{ij}) = G_{TX}(\theta_{ij}, \phi_{ij}, f_{ij}) + G_{RX}(\theta_{ji}, \phi_{ji}, f_{ij}) - L_{AMP}(f_{ij}) - TL(l_{ij}, f_{ij}); \tag{10}$$

$$\theta_{ij} = \arcsin \frac{\overrightarrow{\mathbf{n}'}_i \circ \overrightarrow{P_i P_j}}{\|\overrightarrow{P_i P_j}\|}; \tag{11}$$

$$\phi_{ij} = \arccos \frac{\widehat{\overrightarrow{P_i P_j}} \circ \overrightarrow{\mathbf{v}}_i}{\|\widehat{\overrightarrow{P_i P_j}}\| \cdot \|\overrightarrow{\mathbf{v}}_i\|}. \tag{12}$$

Note that $N_0 = \int_{f_L}^{f_U} psd_{N_0}(f, w)df$ is the ambient noise, where $psd_{N_0}(f, w)$ is the empirical noise power spectral density (psd) for frequency band $[f_L, f_U]$ and w [m/s] is the surface wind speed as in [35]. t_{TTL} is the remaining Time-To-Live (TTL) for the packet, $L_{AMP}(f_{ij})$ [dB] is the power loss of the power amplifier at f_{ij} and $PER_{\max}^{e2e}$ is the maximum e2e error rate for packet m. In these relationships, Eq. (5) is the time after waiting Δt; Eq. (6) calculates the remaining TTL for message m; Eq. (7) calculates the total message's length; Eq. (8) estimates the number of hops $\hat{N}_{hop}^{(i,j)}(t)$ to reach destination d; Eq. (9) estimates the SINR at j while Eq. (10) estimates the total transmission gain in dB from i to j, including the transducer gain at the transmitter and receiver, loss at the power amplifier, and transmission loss; Eqs. (11) and (12) estimate the transducer's radiation angles of j with respect to i. The constraints for $\mathbf{P(i, d, t_{now}}, \mathbf{\Delta t_p})$ are,

(class-independent constraints)

$$P_{TX}^{(i,j)}(t) \geq \int_{(x,y,z)\in\mathcal{U}_{ij}} P_{RX}(i,j,x,y,z)\cdot 10^{-G_{ij}(l_{ij}(t),f_{ij})/10}\cdot g_R(x,y)\cdot g_H(z)dxdydz; \tag{13}$$

$$P_{RX}(i,j,x,y,z) \geq P_{TH}; \tag{14}$$

$$0 \leq \Delta t \leq \frac{t_{TTL}}{\hat{N}_{TX}^{(i,j)}(t)\cdot \hat{N}_{hop}^{(j,d)}(t)}. \tag{15}$$

In these constraints, $P_{RX}(i,j,x,y,z)$ is the received signal power at the generic 3D location (x,y,z) when i transmits to j. Last, $g_R(x,y)$ and $g_H(z)$ are the pdfs of the glider's position on the horizontal plane (i.e., χ-distribution with degree of $2N-2$) and on the vertical direction (i.e., Student's t-distribution with $N-1$ degrees of freedom), respectively [9], P_{TH} is the received power threshold so that the packet can be received with a certain predefined probability. Equation (13) estimates the lower bound of the transmission power to cover the external-uncertainty region so that the received power is above a pre-specified threshold, as accounted for in Eqs. (14) and (15) estimates the bounds of Δt, which must be less than the maximum tolerable delay at the current hop. To support the two classes of delay-tolerant traffic, we have the following additional constraints,

(additional class-dependent constraints)

$$\textbf{Class I} = \begin{cases} \hat{N}_{TX}^{(i,j)}(t) = 1 \\ 1-\left[1-PER(SINR_{ij}(t),\xi)\right]^{\hat{N}_{hop}^{(j,d)}(t)} \leq PER_{\max}^{e2e} \end{cases}; \tag{16}$$

$$\textbf{Class II} = \left\{ \hat{N}_{TX}^{(i,j)}(t) = \left[1-PER(SINR_{ij}(t),\xi)\right]^{-1}. \right. \tag{17}$$

The first constraint for Class I traffic forces packet m to be transmitted only once, while the second constraint guarantees the e2e PER of m should be less than a specified threshold $PER_{\max}^{e2e}$. The constraint for Class II traffic guarantees message m will be transmitted for the average number of times for successful reception at j. By solving the local optimization problem every time when the inputs change significantly (not every time when a packet needs to be sent), i is able to select the optimal next hop j^* so that message m is routed (using minimum network energy) to the external-uncertainty region $\mathcal{U}_{id}$ where destination d should be. Obviously different objective functions (e2e delay, delivery ratio, throughput) could be used depending on the traffic class and mission QoS requirements. Note that in fact our solution can be extended to serve two other classes of traffic—(1) delay-sensitive, loss-tolerant traffic, and (2) delay-sensitive, loss-sensitive traffic—by setting Δt to 0.

To reduce the complexity, we can convert $\mathbf{P(i, d, t_{now}, \Delta t_p)}$ into a discrete optimization problem by considering finite sets of $P_{TX}^{(i,j)}$ and Δt, which can be taken to be a number of equally spaced values within their respective ranges. The problem

then can be solved by comparing the e2e energy consumption estimates of different combination of these discrete values. Assuming that transmission power and time are discretized into N_P and N_{time} values, respectively, for the case of WHOI modem (3 frequencies and 14 combinations of packet types and number of frames [9]), the processor in node i needs to calculate the objective value $42N_P \cdot N_{time} \cdot |\mathcal{N}_i|$ times in each round. The embedded Gumstix motherboard (400 MHz processor and 64 MB RAM) attached to the Micro-Modem is adequate to solve such a problem. To further reduce the computation, instead of running the solution for every packet, it will be rerun only at $t_{now} + \Delta t_p$ for the same class of traffic flow that is sent from i to the same destination d. Here, Δt_p is taken as the minimum of the Δt values of the packets belonging to the same class of traffic and the same destination, estimated from the previous run. Figure 8 depicts an example of how $\mathbf{P(i, d, t_{now}, \Delta t_p)}$ is solved at i. At time t_{now}, the problem is solved with j found to be the next hop to d. The minimum of the Δt values of these packets belonging to the same class of traffic and the same destination observed before t_{now} is $\Delta t'_p$. Packets for d will then be forwarded to j with the calculated transmission power at the selected frequency band until $t_{now} + \Delta t'_p$. Then, the problem is solved again and k is found to be the next hop. The minimum Δt observed so far is $\Delta t''_p$ and, hence, the problem will be solved at $t_{now} + \Delta t'_p + \Delta t''_p$.

Once the optimal frequency band is selected, i needs to notify j to switch to the selected band. A simple protocol can be used as follows. All AUVs use the same frequency band as the Common Control Channel (CCC) to tell the receiver which band is selected. A short packet or preamble with the selected band number is first sent by the transmitter using the CCC, followed by the data packet using selected frequency band after the time for the transmitter and receiver to finish frequency band switching. The receiver will first listen on the CCC, switch to the selected band embedded in the short control packet or preamble, receive the data packet, and then send back a short ACK packet to acknowledge the reception. Finally, both sides switch back to the CCC if the transmission succeeds or the transmission times out. More

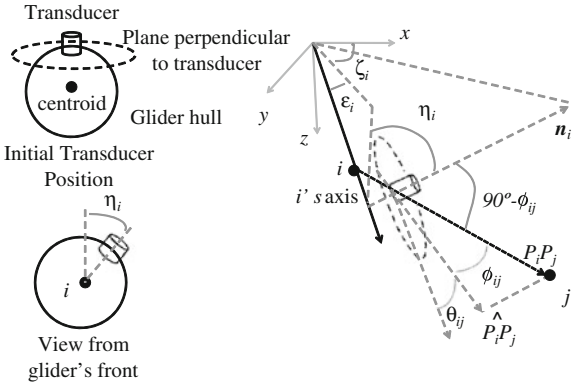


Fig. 8 Derivation of transducer angles from glider i to j

sophisticated frequency-band switching protocols, which are out of the scope of this chapter, can be designed to improve network performance. We rely on the Medium Access Control (MAC) scheme with the WHOI modem to send the data. Since the speed of acoustic wave underwater is very slow when compared with radio waves, the propagation delay has to be considered in order to avoid packet collisions. However, it is difficult to estimate the propagation delay since the positions are uncertain. It may not improve the performance much as the actual propagation delay may be different from the estimation. Moreover, the inter-vehicle traffic underwater is generally low. So the problem of packet collisions is not severe and hence we can just use the MAC scheme provided by the WHOI modem.

6 Performance Evaluation

The communication solution is implemented and tested on our underwater communication emulator [9] as shown in Fig. 9. This underwater acoustic network emulator is composed of four WHOI Micro-Modems [13] and a real-time audio processing card to emulate underwater channel propagation. The multi-input multi-output audio interface can process real-time signals to adjust the acoustic signal gains, to introduce propagation delay, to mix the interfering signals, and to add ambient/man-made noise and interference. Due to the limited number of Micro-Modems (four in our case) and audio processing channels, we can only mix signals from up to three transmitters at the receiver modem (one modem as the receiver and the other three as the transmitters). Therefore, we calculate, select for transmission, and mix with ambient noise, only the three most powerful signals the receiver will encounter. We leave the simulation of more than three simultaneously transmitted signals as a problem for further research.

We are interested in evaluating the performance of the proposed solution in terms of e2e energy consumption, e2e reliability (i.e., e2e delivery ratio), average bit rate

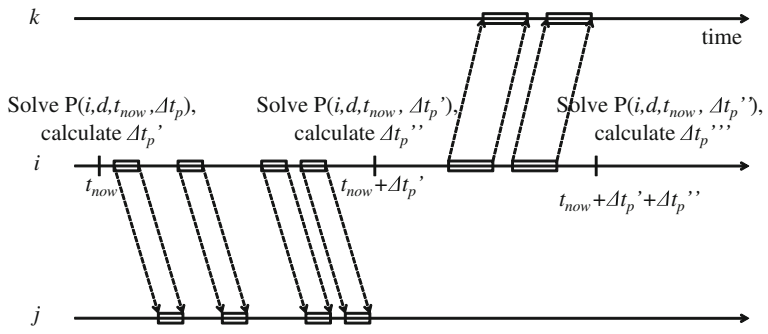


Fig. 9 Solving **P(i, d, t_{now}, Δt_p)** every Δt_p at i

Table 1 Emulation scenario parameters

Parameter	Value
Deployment 3D region	2500 (L) × 2500 (W) × 1000 (H) m^3
Confidence parameter α	0.05
$[P_{min}, P_{max}]$	[1, 10] W
Packet types Ξ	{0, 2, 3, 5}
Glider horizontal speed	0.3 m/s
Gliding depth range	[0, 100] m
Carrier frequencies	10, 15, 25 kHz
$B_{\max}$	10 h

of a link, and overhead, under an environment that is described by the Bellhop model (and the Munk acoustic speed profile in Fig. 4 as input).

Assume that a glider's drifting (i.e., the relative displacement from the glider's trajectory) is a 3D random process $\{X(t), t \geq 0\}$ as the following [30]: (1) In the beginning of the deployment, the drifting is 0, i.e., $X(0) = (0, 0, 0)$; (2) The drifting has independent increments, in that for all $0 \leq t_1 < t_2 < \cdots < t_n$, $X(t_n) - X(t_{n-1})$, $X(t_{n-1}) - X(t_{n-1}), \ldots, X(t_2) - X(t_1)$, $X(t_1)$ are independent; (3) The drifting has stationary increments, in that the distribution of $X(t + s) - X(t)$ does not depend on t and is normally distributed with zero mean and covariance matrix $s\sigma^2 I_3$, where I_3 is the 3×3 identity matrix, and σ is a scaling factor that decides the magnitude of drifting. Note that this drifting model is ideal since the drifting in any of the x, y, z directions is Gaussian. The consideration of realistic drifting pattern is left as future work. Emulation parameters are listed in Table 1. The radiation pattern of the BT-25UF transducer (Fig. 10) is used in the emulations. Every 10 s, a packet is generated in each node. A glider is randomly selected as the collector and half

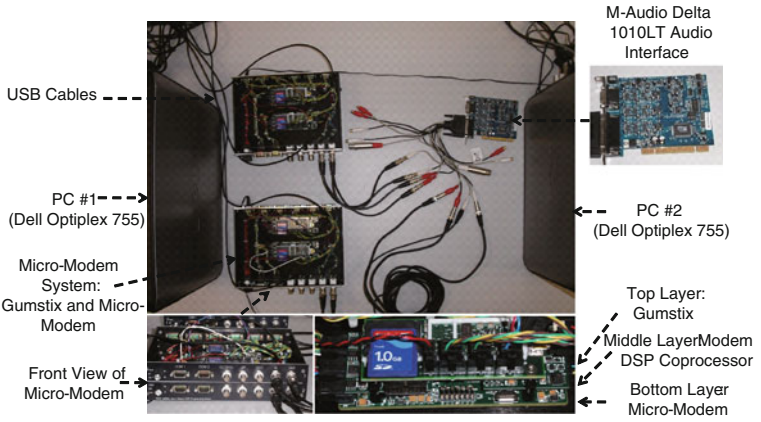


Fig. 10 Underwater communication emulator using WHOI micro-modems

of the other gliders are randomly selected to forward their packets towards it. For statistical relevance, emulations are run for 50 rounds and the average is plotted with 95 % confidence interval. Note that it actually is a scenario for deep water. We will also evaluate the performance in shallow water, where acoustic waves propagate differently.

We are interested in evaluating the performance of our solution for the two classes of traffic in Sect. 5.2, using either the BT-25UF transducer or an ideal omni-directional transducer (with gain equal to 0 dBi). We also want to compare the performance of our solution, which delays the transmission for optimal topology configuration, with the solution without delaying the transmission. For convenience, we denote QUO VADIS for Class I traffic using the BT-25UF transducer, for Class I traffic using the ideal omni-directional transducer, for Class II traffic using the BT-25UF transducer, for Class I traffic using the ideal omni-directional transducer, the solution with no delaying of the transmission (i.e., $\Delta t = 0$ for $\mathbf{P(i, d, t_{now}, \Delta t_p)}$) by 'QUO VADIS I', 'QUO VADIS I - OMNI', 'QUO VADIS II', 'QUO VADIS II-OMNI', and 'QUO VADIS-ND'. We will also compare the performance of our solution (which is closely related to geographic routing and delay-tolerant networking) with geographic routing solutions—MFR, GRS, CRM, and PTKF—and DTN solutions—RAPID, Spray and Wait, and MaxProp—as reviewed in Sect. 3. To make the comparison fair, we use two variant protocols for each of these solutions by adding the constraints of the two classes of traffic to these solution. For example, we denote the MFR solution with Class I constraints in Eq. (16) by 'MFR I', and the solution with Class II constraints in Eq. (17) by 'MFR II'.

The following networking metrics are compared:

- **e2e energy consumption**: the average energy consumed to route one bit of data to the destination;
- **e2e delivery ratio**: the number of data packets received correctly over the number of data packets sent;
- **link bit rate**: the average bit rate between a transmission pair;
- **overhead**: the number of bytes used for position and control to facilitate the transmission of payload data.

Emulations are done for different settings and the results are plotted with 95 % confidence interval and discussed in the following sections.

6.1 Comparison with Geographic Routing Protocols

Our proposed solution forwards packets based on the geographic location. To see how well our solution performs against existing geographic routing protocols, simulations are run and the results are plotted in Figs. 11 and 12 (as existing research on underwater geographic routing is still very limited, the solutions we compare against are taken from those originally designed for terrestrial wireless networks). As shown in these two figures, we can see that QUO VADIS has better performance

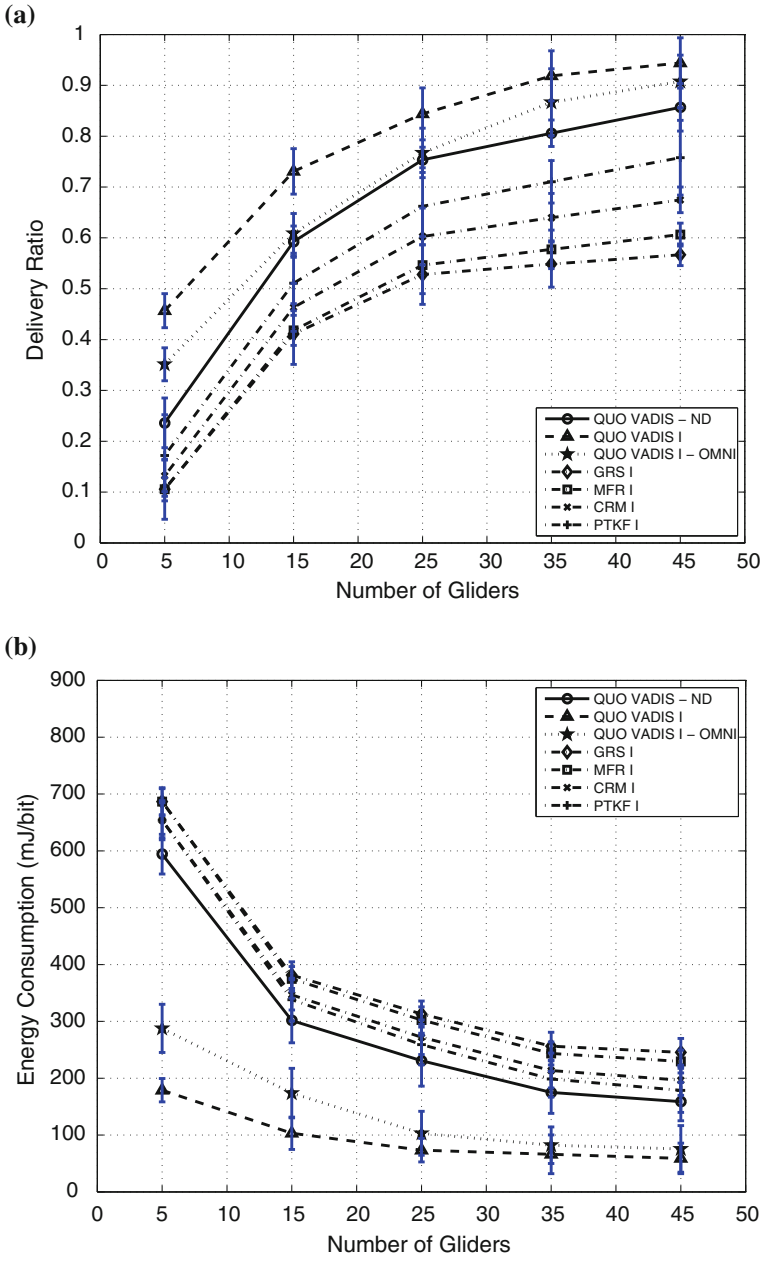


Fig. 11 Performance comparison for Class I traffic with *geographic routing* protocols. **a** Delivery ratio comparison. **b** Energy consumption comparison. **c** Link bit rate comparison

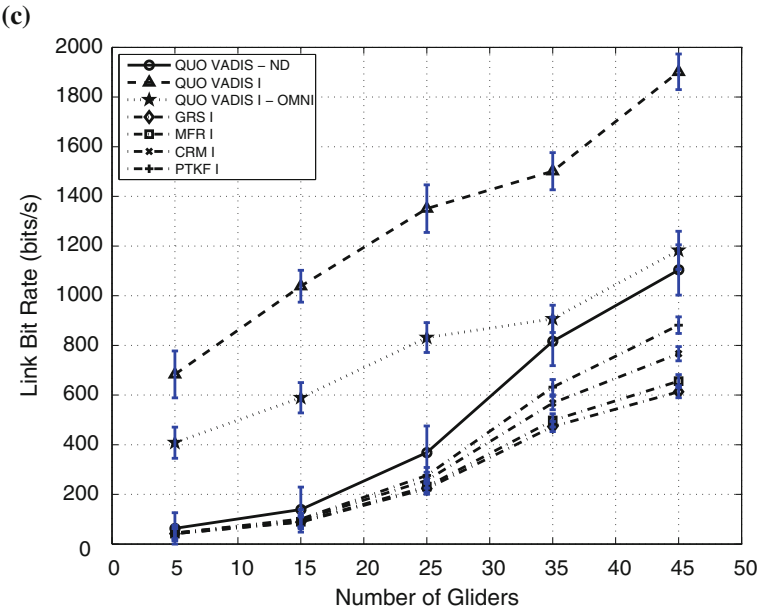


Fig. 11 (continued)

than QUO VADIS-OMNI and QUO VADIS-ND for the same class of traffic in terms of these three metrics. By delaying packet transmissions to wait for the optimal network topology, the e2e energy consumption is reduced while the e2e delivery ratio and link bit rate increase (e.g., with 5 gliders, the energy consumption for QUO VADIS I is around 30 % of that for QUO VADIS-ND). By exploiting the frequency-dependent radiation pattern of the transducer, received signal power may obtained a gain of up to 20 dB, which we observed in the simulations. Hence QUO VADIS using the BT-25UF transducer has better performance than that using the omni-directional transducer. Due to the QoS requirements, retransmissions are needed to recover link errors, resulting in higher e2e delivery ration for Class II traffic than for Class I traffic. On the other hand, this leads to more energy consumption.

Different versions of our QUO VADIS solutions also perform better than geographic routing protocols GRS, MFR, CRM, and PKTF. This is because that the uncertainty in location leads to errors in route selection, packet transmissions, and transmission power estimates. Also these geographic routing protocols do not consider the propagation delay underwater, which results in degraded communication performance. Interesting enough, we can see that among these geographic routing protocols, PKTF offers the best performance. This is because it jointly considers the transmission power and routing to minimize the e2e energy consumption. Therefore it performs better than the other geographic routing protocol, which only consider the distance or angle metrics for routing (not closely related to network performance). GRS gives the worst performance since it generally needs to forward the packet to

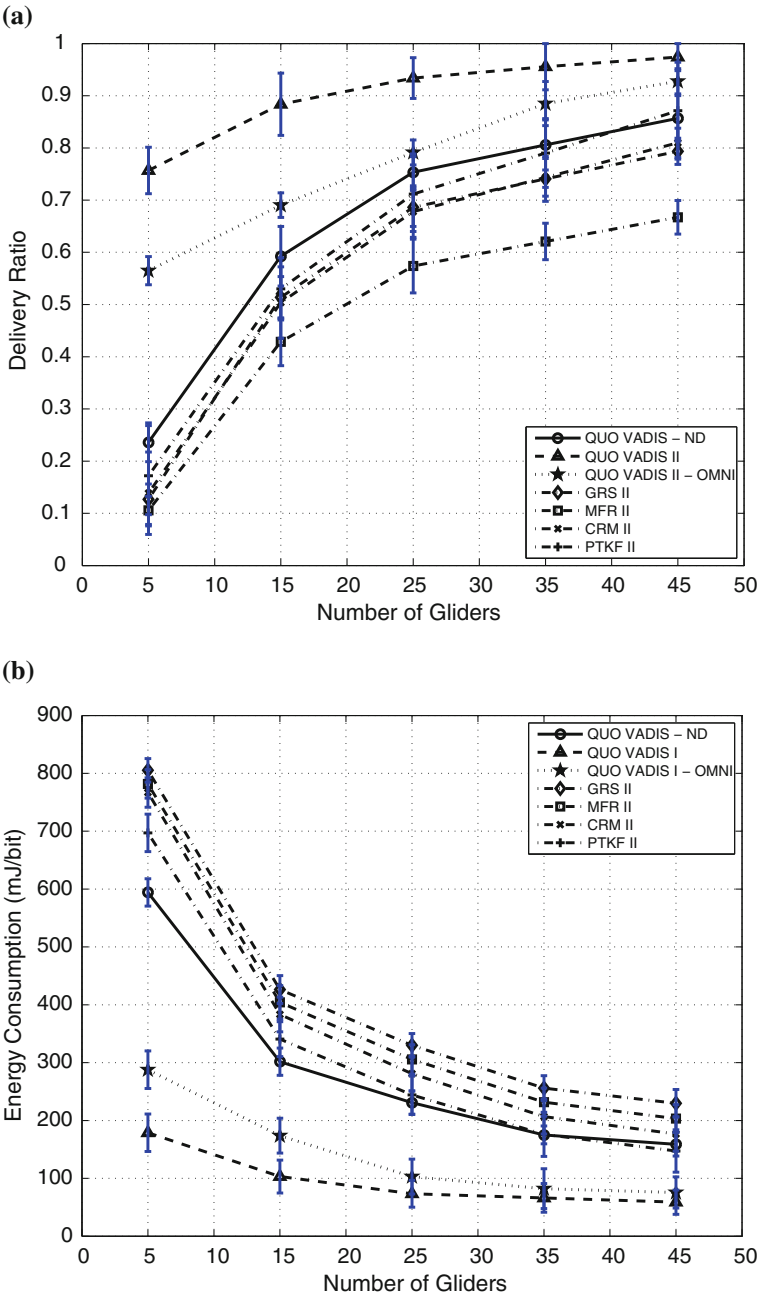


Fig. 12 Performance comparison for Class II traffic with *geographic routing* protocols. **a** Delivery ratio comparison. **b** Energy consumption comparison. **c** Link bit rate comparison

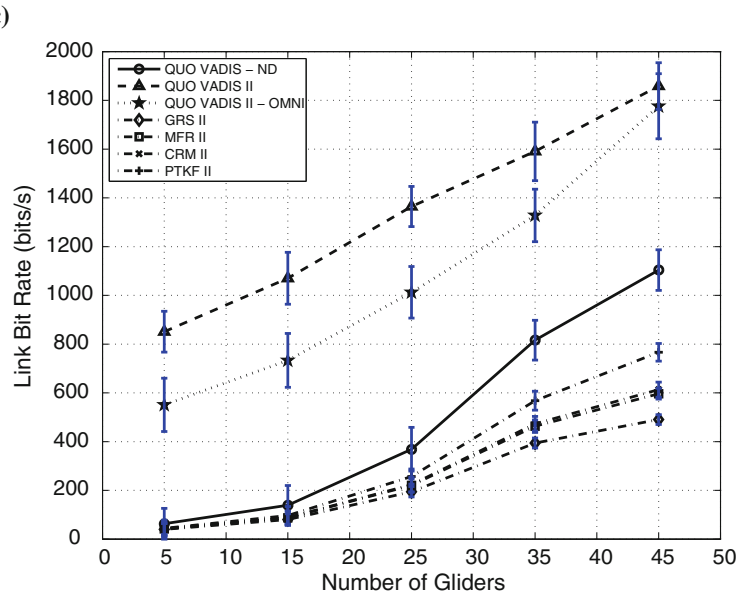


Fig. 12 (continued)

the node that is far from the transmitter, which introduces bad link performance. Similarly, CRM performs better than MFR as the CRM has less probability to forward packets to node that is far away than MFR does.

6.2 Comparison with DTN Solutions

Similar to the comparison against the geographic routing solutions, we compare the performance of QUO VADIS against the DTN solutions—RAPID, MaxProp and Spray and Wait. As shown in Figs. 13 and 14, QUO VADIS gives improved performance over RAPID, MaxProp and Spray and Wait. That is mainly due to that these DTN solutions transfer packets once the neighbors are in the transmission range. Such schemes may be good for scenarios where the connectivity is intermittent. However, the performance may not be optimal since this may not be the time to achieve the best link performance. In contrast, QUO VADIS predicts and waits for the best network configuration, where nodes move closer for the best communications. So the e2e delivery ratio and link bit rate of QUO VADIS is the highest while its energy consumption is minimal. Note that among these compared DTN solutions, RAPID performs the best. This is because RAPID prioritizes old packets so they won't be dropped. MaxProp gives priority to new packets; older, undelivered packets will be dropped in the middle. Spray and Wait works in a similar way, which does not give

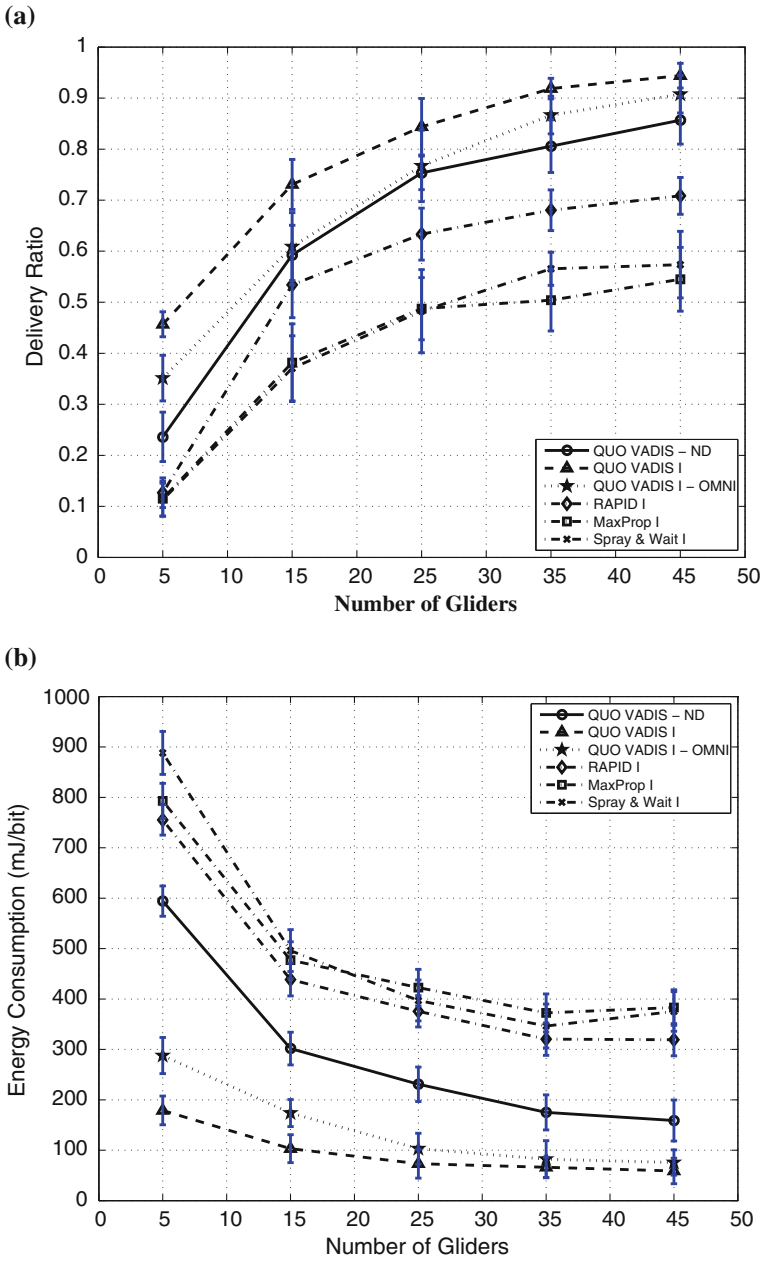


Fig. 13 Performance comparison for Class I traffic with *DTN* protocols. **a** Delivery ratio comparison. **b** Energy consumption comparison. **c** Link bit rate comparison

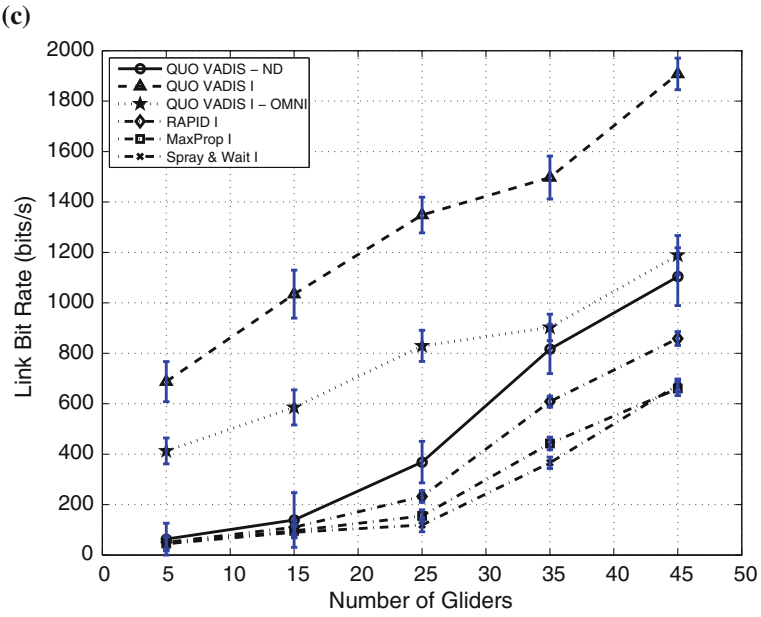


Fig. 13 (continued)

priority to older packets. On the other hand, Spray and Wait is slightly better than MaxProp. This is because in our scenario, the network connectivity is not disrupt. The way that MaxProp routes (based on the e2e delivery ratio estimation) will be very different from that Spray and Wait does (i.e., just transmits the packet to a neighbor then lets the neighbor continue to forward it). Moreover, MaxProp still needs to pay for the overhead to obtain the global e2e delivery ratio information.

6.3 Overhead Comparison

We plot and compare the overheads (per node) of these protocols in Fig. 15. Note that as QUO VADIS, QUO VADIS-ND, and QUO VADIS-OMNI work almost the same way, i.e., the uncertainty region information is broadcast periodically (here the period is taken to be 60 s), their overheads are the same and thus we use QUO VADIS in the figure to represent these variant versions. Similarly, nodes running the geographic routing protocols GRS, MFR and CRM only need to periodically broadcast the position information so their overhead is basically the same. Hence we use GRS/MFR/CRM to represent them.

Surprisingly, even though QUO VADIS achieves the best network performance, its overhead is not the biggest. The protocols with the larger overhead are RAPID and MaxProp. In order to work, RAPID needs the following control information: average

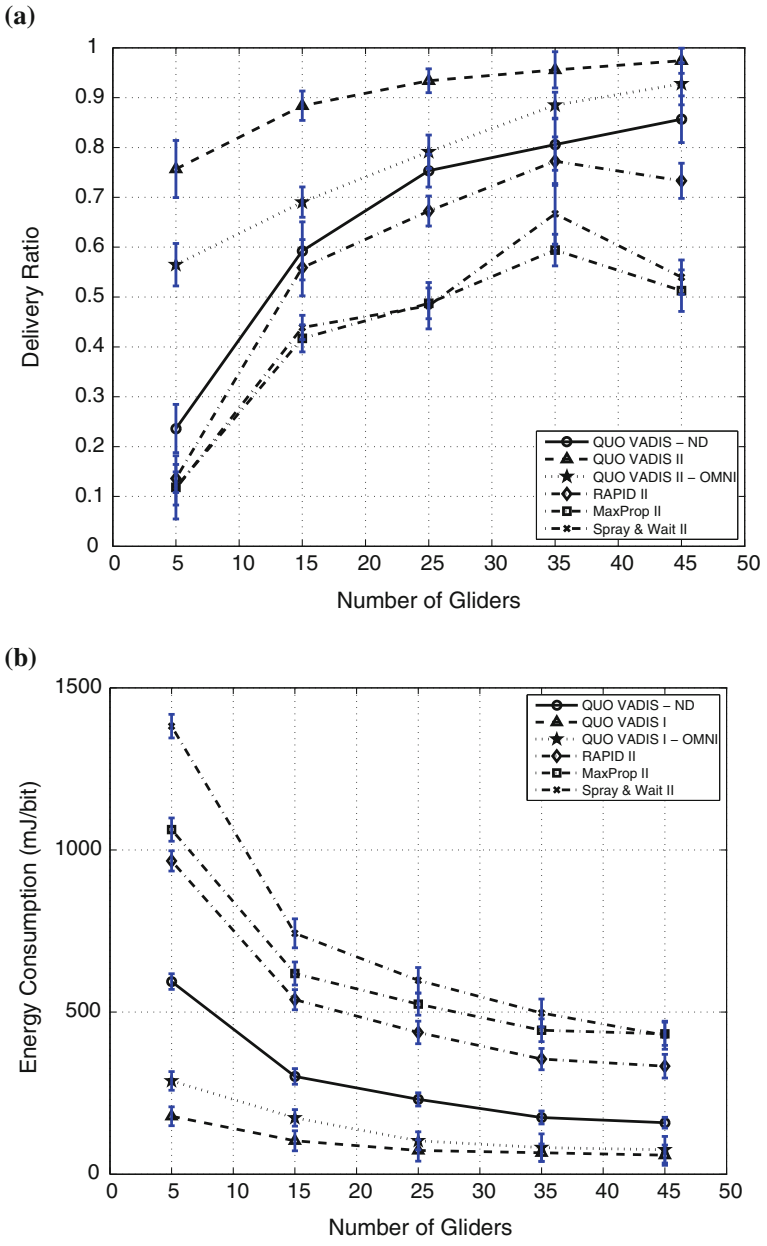


Fig. 14 Performance comparison for Class II traffic with *DTN* protocols. **a** Delivery ratio comparison. **b** Energy consumption comparison. **c** Link bit rate comparison

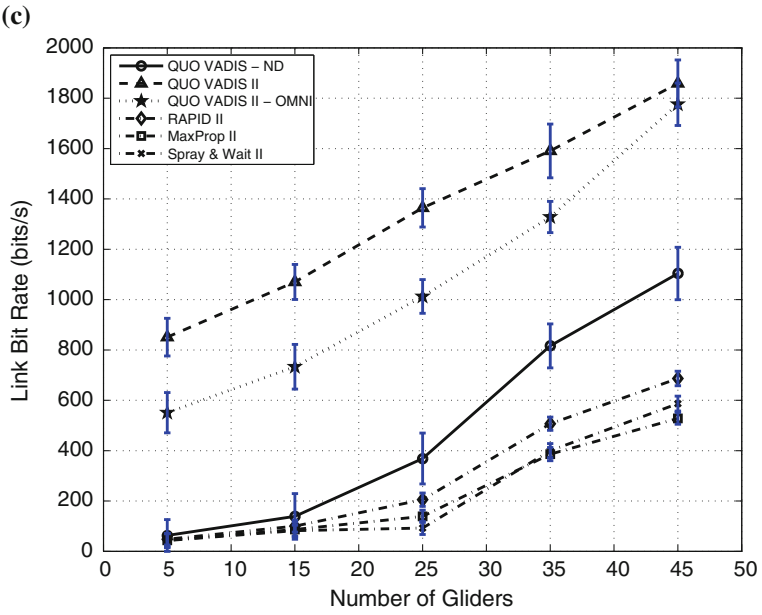


Fig. 14 (continued)

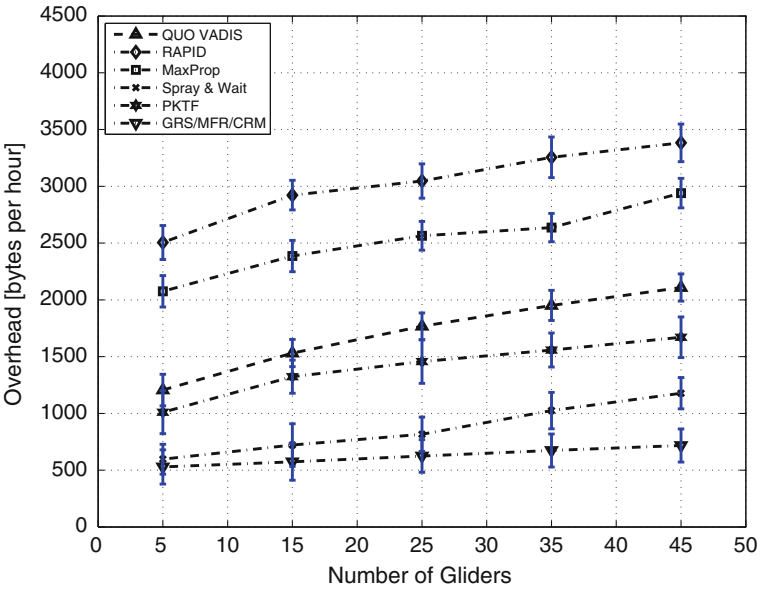


Fig. 15 Comparison of the overhead

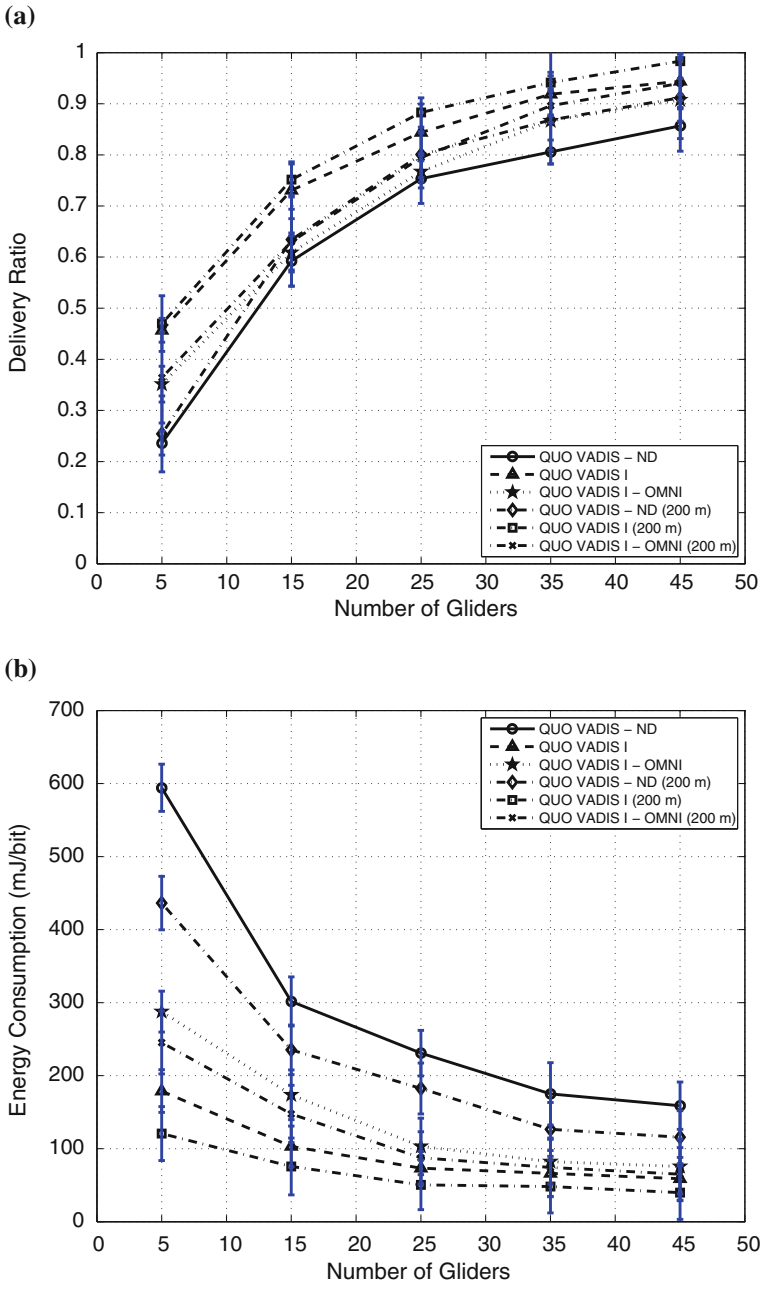


Fig. 16 *Shallow water*: performance comparison for Class I traffic. **a** Delivery ratio comparison. **b** Energy consumption comparison. **c** Link bit rate comparison

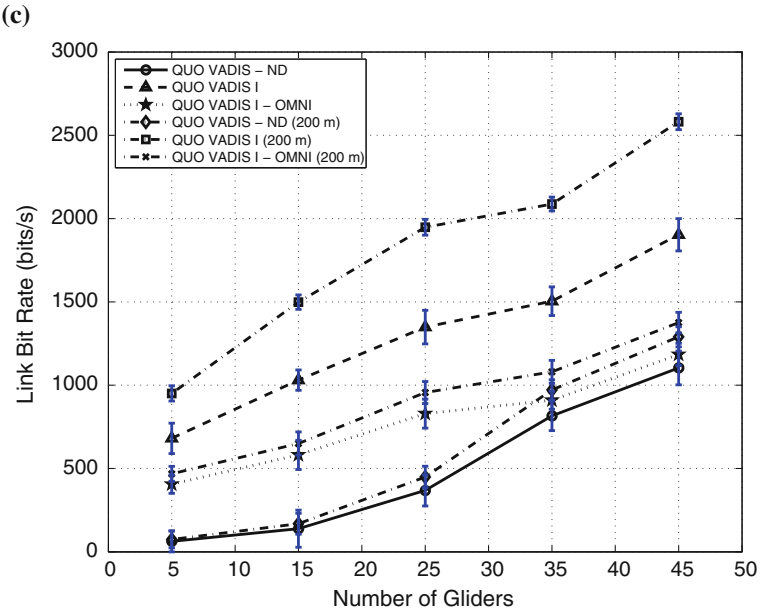


Fig. 16 (continued)

size of past transfer opportunities, expected meeting times with nodes, list of packets delivered since last exchange, the updated delivery delay estimate based on current buffer state, and information about other packets if modified since last exchange with the peer, which takes a large number of bytes. MaxProp needs to exchange a list of the probabilities of meeting every other node on each contact, which is basically global information. It also has the neighbor discovery overhead. Compared to RAPID and MaxProp, QUO VADIS only needs to exchange the external uncertainty information of itself and the destination node, which is obviously less. On the other hand, PKTF needs a probe message that has five data fields. Only the nodes in the selected path are required to respond with a probe—whether it is sent for the forwarding or reverse direction. The Spray and Wait protocol reduces transmission overhead by spreading only a few number of data packets to the neighbors. The source node then stops forwarding and lets each node carrying a copy perform direct transmission. In our emulation, we select the number to be one to make the comparison fair and hence the overhead is small. Lastly, for the other geographic routing protocols GRS, MFR, and CRM, the nodes just need to know the geographic locations of the neighbors and the destination. Therefore the overhead required is the least. Note that here it is not necessary to differentiate the two classes of traffic since the overhead difference is small.

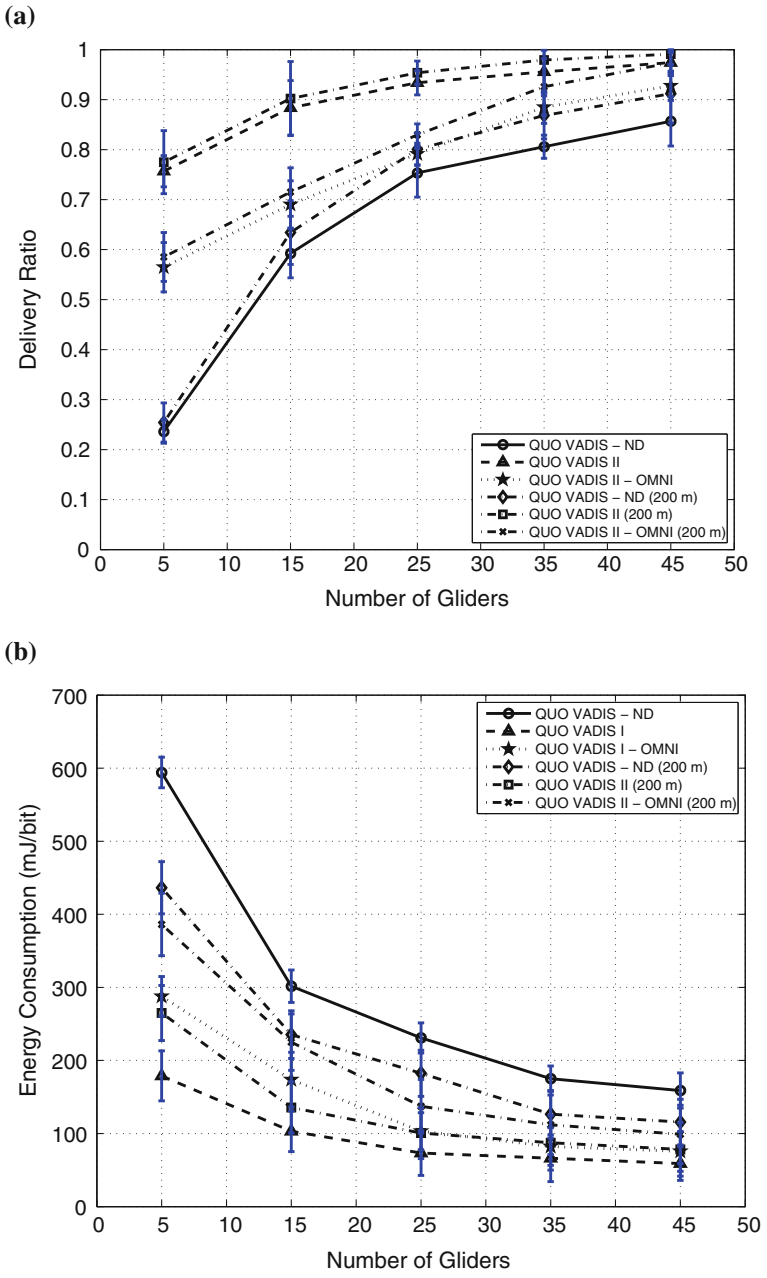


Fig. 17 *Shallow water*: performance comparison for Class II traffic. **a** Delivery ratio comparison. **b** Energy consumption comparison. **c** Link bit rate comparison

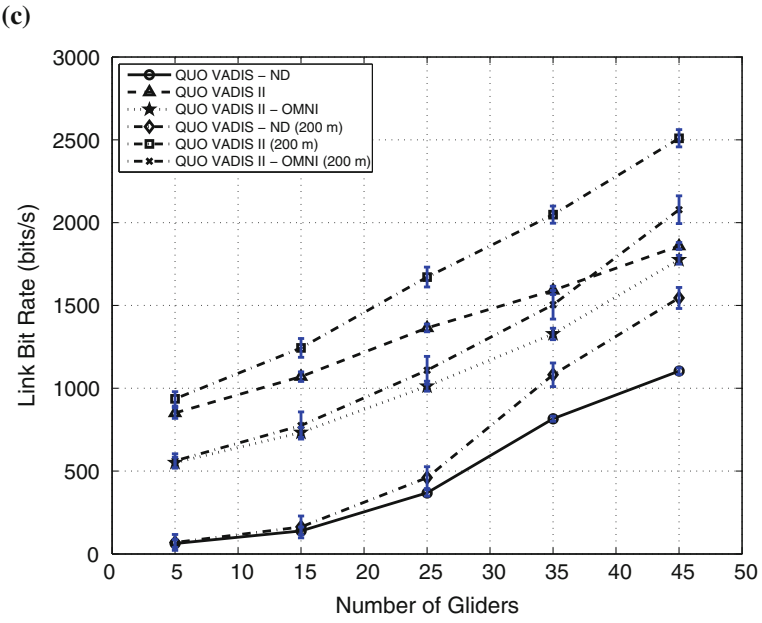


Fig. 17 (continued)

6.4 Performance in Shallow Water

So far the results are obtained using the setting in Table 1, which is for the deep water. We change the network scenario to the shallow water scenario by setting the depth of the 3D region to 200 m. Note that generally there is no definite depth value for shallow water as the sound propagation depends on the corresponding underwater environment. Therefore in some references (e.g., [1]), the shallow water is considered to be less than 100 m deep, while for some other acoustic researchers this depth can be up to 500 m [20]. Here we use 200 m, which is used by National Oceanic and Atmospheric Administration (NOAA) as the depth for the first layer (and this depth is used in quite a few cases in the well-known set of benchmark shallow water test cases presented in the Shallow Water Acoustic Modeling Workshop 1999 (SWAM'99) [31]). In this shallow water scenario, the path loss estimated by the Urick's model is very different from that estimated by the Bellhop model. We had anticipated the performance will degrade because of this mismatch. Surprising enough, as shown in Figs. 16 and 17, we find the performance (in terms of e2e delivery ratio, energy consumption, and link bit rate) in the shallow water is actually better. A more careful analysis reveals the reason—the existence of the *surface duct* in the shallow water. Surface duct is basically a zone below the sea surface where sound rays are refracted toward the surface and then reflected. The rays alternately are refracted and reflected along the duct out to relatively long distances from the sound source. Hence the

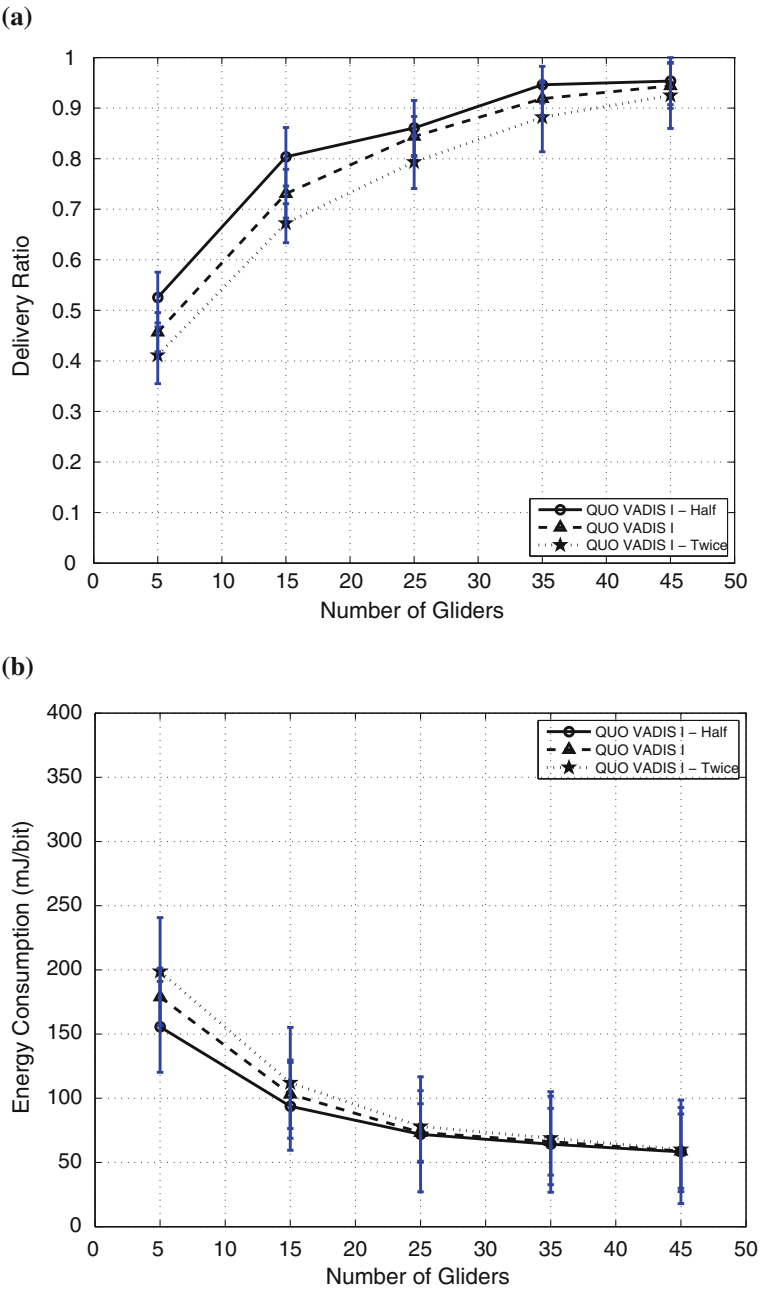


Fig. 18 *Uncertainty update interval*: performance comparison for Class I traffic. **a** Delivery ratio comparison. **b** Energy consumption comparison. **c** Link bit rate comparison

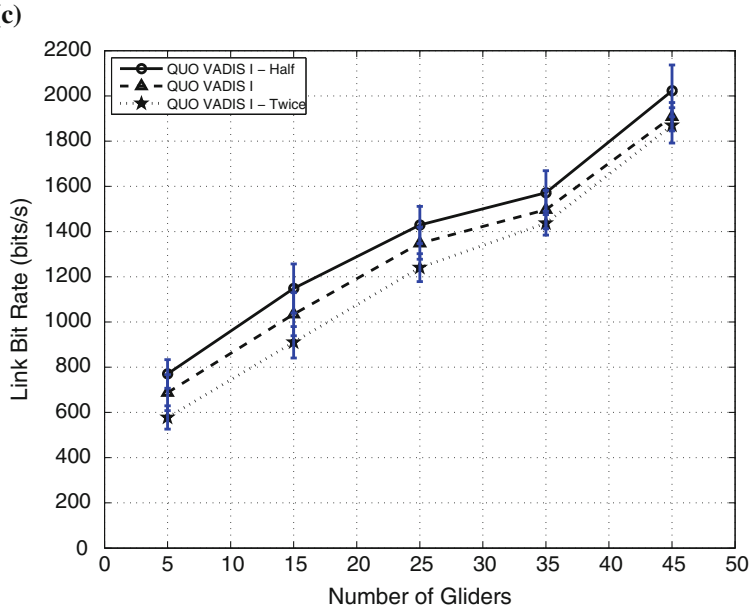


Fig. 18 (continued)

acoustic waves are relatively concentrated in the surface duct, leading to less path loss. This consequently leads to improved network performance.

6.5 Performance Using Different Uncertainty Update Intervals

So far the broadcast interval of uncertainty region is fixed to 60 s. Our last interest is to evaluate the performance of the QUO VADIS variants when different broadcast intervals are used. Therefore we re-run the emulations for two more cases: (i) half of interval (i.e., 30 s); and (ii) double of interval (i.e., 120 s). From Figs. 18 and 19, we can see that the performance of the QUO VADIS variants becomes worse when the update interval is doubled. This is because when the interval is doubled, the position uncertainty information becomes less accurate. This leads to larger error in neighbor selection for packet forwarding and in the estimation of transmission power. On the other hand, halving the interval leads to performance improvement as the uncertainty information is updated in a more timely manner (therefore routing error becomes smaller and transmission power is better estimated). However, this obviously leads to the increase in overhead. Therefore the tradeoff between overhead and metrics such as delivery ratio, energy consumption, and link bit rate should be carefully considered for different applications. Here we use "QUO VADIS-Half," "QUO VADIS," and

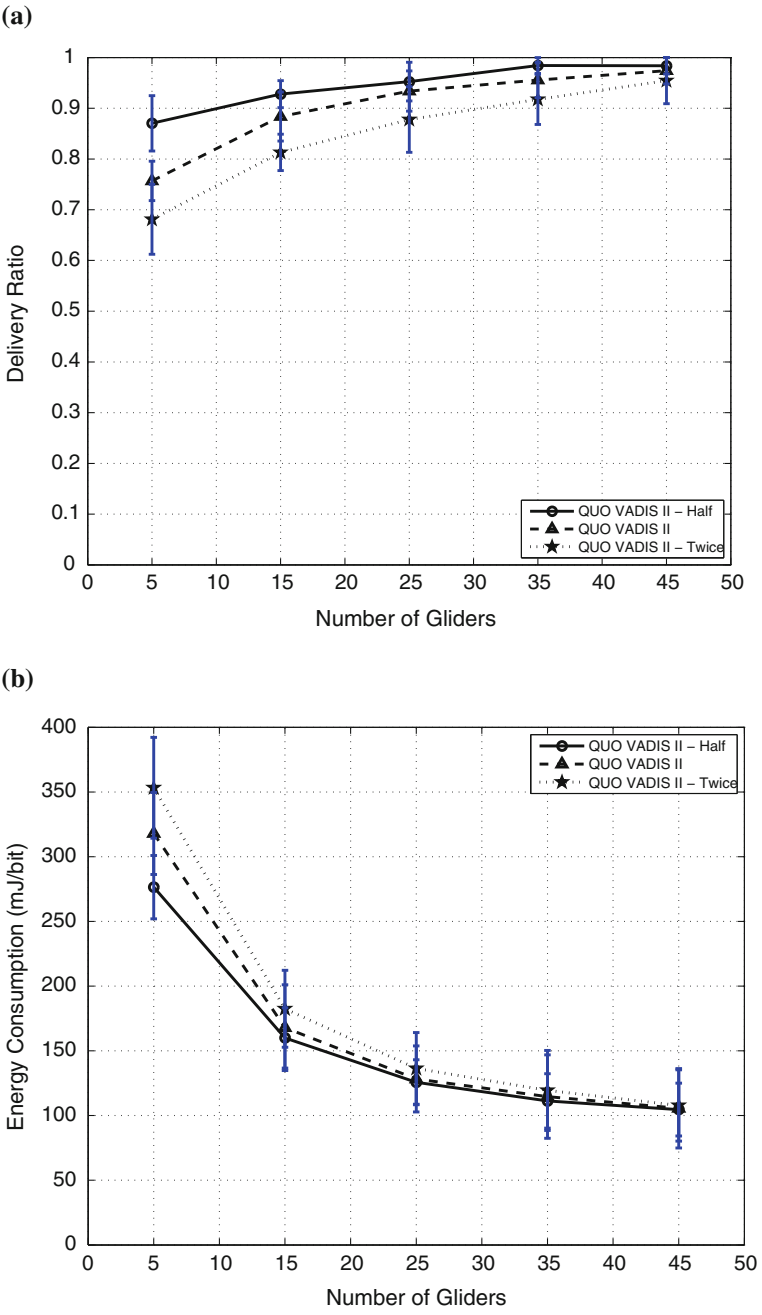


Fig. 19 *Uncertainty update interval*: performance comparison for Class II traffic. **a** Delivery ratio comparison. **b** Energy consumption comparison. **c** Link bit rate comparison

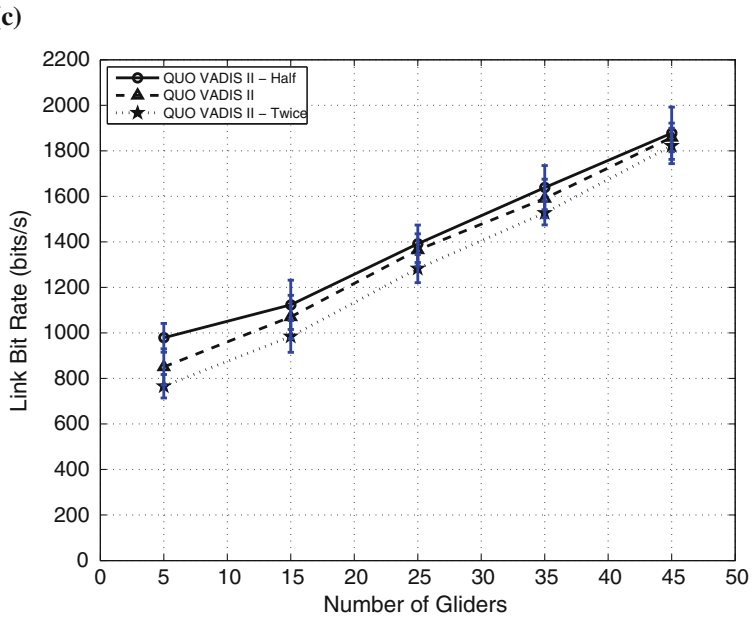


Fig. 19 (continued)

"QUO VADIS-Twice" to denote the cases with update interval of 30, 60, and 120 s, respectively.

To sum up, our proposed framework QUO VADIS improves the network performance for delay-tolerant applications in terms of e2e energy consumption, delivery ratio, and link bit rate by waiting for a 'favorable' topology configuration and by exploiting the gains of directional transducers. Through emulations for different setups, we demonstrated that they can offer better performance than the well-known geographic routing and DTN protocols when serving two classes of delay-tolerant traffic.

7 Conclusion

We proposed QUO VADIS, a **Q**oS-aware **u**nderwater **o**ptimization framework for inter-**v**ehicle communication using **a**coustic **di**rectional transducer**s**. Based on the trajectory and position uncertainties of the AUVs, an AUV predicts a favorable network topology with relatively short links in the future and postpones transmission in favor of a lower transmission energy and a higher data rate. Communication energy consumption is further reduced by exploiting the frequency-dependent radiation pattern of underwater acoustic transducers. The proposed solution is implemented and tested in our underwater communication emulator, showing improvement over some

well-known geographic routing protocols and DTN protocols in terms of e2e energy consumption, reliability, and link bit rate.

References

1. I.F. Akyildiz, D. Pompili, T. Melodia, Underwater acoustic sensor networks: research challenges. Ad Hoc Netw. (Elsevier) **3**(3), 257–279 (2005)
2. A. Balasubramanian, B. Levine, A. Venkataramani, DTN routing as a resource allocation problem, in *Proceedings of ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM)*, 2007
3. M. Blain, S. Lemieux, R. Houde, Implementation of a ROV navigation system using acoustic/Doppler sensors and Kalman filtering, in *Proceedings of IEEE International Conference on Engineering in the Ocean Environment (OCEANS)*, San Diego, 2003
4. J. Burgess, B. Gallagher, D. Jensen, B.N. Levine, MaxProp: routing for vehicle-based disruption-tolerant networks, in *Proceedings of Conference on Computer Communications (INFOCOM)*, Barcelona, 2006
5. S. Burleigh, A. Hooke, L. Torgerson, K. Fall, V. Cerf, B. Durst, K. Scott, H. Weiss, Delay-tolerant networking: an approach to interplanetary Internet. IEEE Commun. Mag. **41**(6), 128–136 (2003)
6. G. Casella, R.L. Berger, *Statistical Inference*, 2nd edn. (Duxbury Press, Pacific Grove, 2001)
7. J. Catipovic, Performance limitations in underwater acoustic telemetry. IEEE J. Oceanic Eng. **15**, 205–216 (1990)
8. C.Y. Chan, M. Motani, An integrated energy efficient data retrieval protocol for underwater delay tolerant networks, in *Proceedings of IEEE International Conference on Engineering in the Ocean Environment (OCEANS)—Europe*, Aberdeen, 2007
9. B. Chen, P. Hickey, D. Pompili, Trajectory-aware communication solution for underwater gliders using WHOI micro-modems, in *Proceedings of IEEE Communications Society Conference on Sensor, Mesh, and Ad Hoc Communications and Networks (SECON)*, Boston, 2010
10. B. Chen, D. Pompili, QUO VADIS: QoS-aware underwater optimization framework for inter-vehicle communication using acoustic directional transducers, in *Proceedings of IEEE Communications Society Conference on Sensor, Mesh, and Ad Hoc Communications and Networks (SECON)*, Salt Lake City, 2011
11. K. Fall, A delay-tolerant network architecture for challenged Internets, in *Proceedings of ACM Special Interest Group on Data Communication (SIGCOMM)*, Karlsruhe, 2003
12. G.G. Finn, *Routing and addressing problems in large metropolitan scale Internetworks*, Technical report, ISI, Storrs, 1987
13. L. Freitag, M. Grund, S. Singh, J. Partan, P. Koski, K. Ball, The WHOI micro-modem: an acoustic communications and navigation system for multiple platforms, in *Proceedings of IEEE International Conference on Engineering in the Ocean Environment (OCEANS)*, Washington DC, 2005
14. S.A.L. Glegg, R. Pirie, A. LaVigne, *A study of ambient noise in shallow water*, Florida Atlantic University Technical Report, 2000
15. Z. Guo, G. Colombi, B. Wang, J.H. Cui, D. Maggiorini, Adaptive routing in underwater delay/disruption tolerant sensor networks, in *Proceedings of IEEE/IFIP Conference on Wireless on Demand Network Systems and Services (WONS)*, Garmisch-Partenkirchen, 2008
16. Z. Guo, B. Wang, J.H. Cui, Prediction assisted single-copy routing in underwater delay tolerant networks, in *Proceedings of IEEE Global Communications Conference (GLOBECOM)*, Miami, 2010
17. D.B. Kilfoyle, A.B. Baggeroer, The state of the art in underwater acoustic telemetry. IEEE J. Oceanic Eng. **25**, 4–27 (2000)

18. E. Kranakis, H. Singh, J. Urrutia, Compass routing on geometric networks, in *Proceedings of Canadian Conference on Computational Geometry*, Vancouver, 1999
19. H. Luo, Z. Guo, W. Dong, F. Hong, Y. Zhao, LDB: localization with directional beacons for sparse 3D underwater acoustic sensor networks. J. Netw. **5**(1), 28–38 (2010)
20. J. Lynch, Acoustical oceanography and shallow water acoustics, in *Proceedings of Australian Acoustical Society Conference (ACOUSTICS)*, 2011
21. E. Magistretti, J. Kong, U. Lee, M. Gerla, P. Bellavista, A. Corradi, A mobile delay-tolerant approach to long-term energy-efficient underwater sensor networking, in *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC)*, Kowloon, 2007
22. T. Melodia, D. Pompili, I.F. Akyildiz, On the interdependence of distributed topology control and geographical routing in ad hoc and sensor networks. IEEE J. Sel. Areas Commun. **23**(3), 520–532 (2005)
23. J.M. Montana, M. Stojanovic, M. Zorzi, Focused beam routing protocol for underwater acoustic networks, in *Proceedings of ACM International Workshop on Underwater Networks (WUWNet)*, San Francisco, 2008
24. J. Partan, J. Kurose, B.N. Levine, A survey of practical issues in underwater networks, in *Proceedings of ACM International Workshop on UnderWater Networks (WUWNet)*, Los Angeles, 2006
25. D. Pompili, I.F. Akyildiz, A cross-layer communication solution for multimedia applications in underwater acoustic sensor networks, in *Proceedings of IEEE International Conference on Mobile Ad-Hoc and Sensor systems (MASS)*, Atlanta, 2008
26. M. Porter, BELLHOP Gaussian Beam/Finite Element Beam Code. http://oalib.hlsresearch.com/Rays/index.html. Accessed 11 Jan 2012
27. J. Proakis, J. Rice, E. Sozer, M. Stojanovic, Shallow water acoustic networks, in *Encyclopedia of Telecommunications*, ed. by J.G. Proakis (Wiley, New York, 2003)
28. J.G. Proakis, E.M. Sozer, J.A. Rice, M. Stojanovic, Shallow water acoustic networks. IEEE Commun. Mag. **39**(11), 114–119 (2001)
29. A. Quazi, W. Konrad, Underwater acoustic communication. IEEE Commun. Mag. **20**, 24–29 (1982)
30. S.M. Ross, *Introduction to Probability Models*, 8th edn. (Academic Press, San Diego, 2003)
31. K.B. Smith, A. Tolstoy, Summary of results for swam'99 test cases, in *Proceedings of a Shallow Water Acoustic Modeling Workshop (SWAM)*, Monterey, 1999
32. T. Spyropoulos, K. Psounis, C.S. Raghavendra, Spray and wait: an efficient routing scheme for intermittently connected mobile networks, in *Proceedings of the ACM SIGCOMM Workshop on Delay-Tolerant Networking (WDTN)*, 2005
33. V. Srivastava, M. Motani, Cross-layer design: a survey and the road ahead. IEEE Commun. Mag. **43**(12), 112–119 (2005)
34. M. Stojanovic, Acoustic (underwater) communications, in *Encyclopedia of Telecommunications*, ed. by J.G. Proakis (Wiley, New York, 2003)
35. M. Stojanovic, On the relationship between capacity and distance in an underwater acoustic communication channel, in *Proceedings of ACM International Workshop on UnderWater Networks (WUWNet)*, Los Angeles, 2006
36. H. Takagi, L. Kleinrock, Optimal transmission ranges for randomly distributed packet radio terminals. IEEE Trans. Commun. **COM-32**(3), 246–257 (1984)
37. R.J. Urick, *Principles of Underwater Sound* (McGraw-Hill, New York, 1983)
38. S. Williams, I. Mahon, Simultaneous localisation and mapping on the Great Barrier Reef, in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, New Orleans, 2004
39. X. Xiang, Z. Zhou, X. Wang, Self-adaptive on demand geographic routing protocols for mobile ad hoc networks, in *Proceedings of IEEE International Conference on Computer Communications (INFOCOM)*, Anchorage, 2007
40. P. Xie, J.H. Cui, L. Lao, VBF: vector-based forwarding protocol for underwater sensor networks, in *Proceedings of IFIP Networking*, Waterloo, 2005

Part VII
Multimedia and Body Sensor Networks

Chapter 14
Low-Complexity Video Streaming for Wireless Multimedia Sensor Networks

Scott Pudlewski and Tommaso Melodia

Abstract In recent years, there has been intense research and considerable progress in solving numerous wireless sensor networking challenges. However, the key problem of enabling real-time quality-aware multimedia transmission over wireless sensor networks is largely unexplored. The large amount of data generated by most multimedia applications (compared to traditional scalar sensor networks), along with the higher QoS requirements make it difficult to meet the low energy use requirements of practical sensor networks. We explore the use of compressed sensing (aka "compressive sampling") to reduce the energy required to encode and transmit high quality video in a severely resource-constrained environment. In this chapter, we will examine some of the major challenges of wireless multimedia sensor network (WMSN) implementation. Specifically, we examine what it would take to develop a WMSN that has similar performance (and restrictions) as a traditional scalar wireless sensor network (WSN). We then examine how we can use the new paradigm of compressed sensing (CS) to solve many of these problems.

1 Introduction

Advances in sensing, computation, storage, and wireless networking are driving an increasing interest in multimedia [1–4] and participatory [5, 6] sensing applications. While these applications show high promise, they require wirelessly networked streaming of video originating from devices that are constrained in terms of

S. Pudlewski (✉)
Lincoln Laboratory, Massachusetts Institute of Technology, Lexington, MA, USA
e-mail: scott.pudlewski@ll.mit.edu

T. Melodia
Department of Electrical Engineering, State University of New York (SUNY), Buffalo, NY, USA
e-mail: tmelodia@eng.buffalo.edu

H. M. Ammari (ed.), *The Art of Wireless Sensor Networks*,
Signals and Communication Technology, DOI: 10.1007/978-3-642-40066-7_14,

instantaneous power, energy storage, memory, and computational capabilities. However, state-of-the-art technology, for the most part based on streaming predictively encoded video (e.g., MPEG-4 Part 2, H.264/AVC [7–9], H.264/SVC [10]) through a layered wireless communication protocol stack, is not appropriate for wireless multimedia sensor networks (WMSNs) because of the following limitations:

- **Predictive Video Encoding is Computationally Intensive.** State-of-the-art predictive encoding requires calculating motion vectors, which is a computationally intensive operation, requires significant power consumption and complexity at the sensor node. Ideally, a WMSN system would transfer most of the complexity to the multimedia sink, which is in general *not* a resource-constrained system.
- **Predictive Encoding of Video Increases Impact of Channel Errors.** In existing layered protocol stacks (e.g., IEEE 802.11 and 802.15.4), frames are split into multiple packets. Any errors in even one of these packets, after a cyclic redundancy check, can cause visible distortion in a video frame. Because of the predictive nature of modern video encoders, distortion can then propagate to tens or even hundreds of frames that are dependent on the distorted frame. *Structure in video representation, which plays a fundamental role in our ability to compress video, is detrimental when it comes to wireless video transmission with lossy links.*

Both of these limitations lead to an increase in complexity at the sensor node. In the first case, the computational complexity of traditional video encoders immediately restricts the type of processors that can be used. There are many commercially available processors that can encode video, but they require much more power, and are much more expensive, than what is appropriate for a WSN.

To compensate for noisy channels, increasing the received signal-to-noise ratio (SNR) is often necessary to reduce the number of bit errors to an acceptable level. Since power is limited in real systems, other methods have been developed to decrease the BER. Traditionally, forward error correction (FEC) methods (i.e., Reed-Solomon [11] codes or RCPC [12] codes) are employed to reduce the BER for a fixed SNR. However, FEC will increase the size of each encoded packet, which could result in a net *increase* in total energy required for transmission. In addition, this will again lead to an increase in complexity.

Compressed sensing (CS) [13–19] is a promising technique for dealing with these limitations. Compressed sensing (aka "compressive sampling") is a new paradigm that allows the faithful recovery of signals from $M << N$ measurements where N is the number of samples required for the Nyquist sampling. Since these M measurements are created by taking M linear combinations of the N pixels, CS can offer an alternative to traditional video encoders by enabling imaging systems that sense and compress data simultaneously *at very low computational complexity for the encoder* [20, 21].

The remainder of this chapter is structured as follows. In Sect. 2 we highlight some of the challenges in developing a video transmissions system for wireless multimedia sensor networks. We introduce compressed sensing in Sect. 3, we introduce compressive imaging in Sect. 4, and we introduce CS-based video encoding in Sect. 5. CS-based error correction ia introduced in Sect. 6. The energy-rate-distortion

analysis of the CS-based video encoder is presented in Sect. 7, and rate control of the CS video is presented in Sect. 8. Finally, future work is presented in Sect. 9 and in Sect. 10 we draw the main conclusions.

2 Challenges

Multimedia networking applications are normally characterized by high complexity and high data rate. However, sensor nodes are ideally low-cost, *low-complexity* devices that have a long network lifetime, which generally leads to a lower data rate than other types of networks. For a practical WMSN implementation, a video encoding system must be designed that can fit within these constraints. Below, we examine some of the key constraints.

2.1 Complexity Constraints

While high-end mobile devices have recently become commercially available (i.e., smartphones, tablets), WMSN sensor nodes should ideally be simple, low-complexity devices. For example, imote2 nodes [22], earlier generation imote nodes, or other WSN platforms are much cheaper and have a much longer battery life than even low-end smartphones. However, this increase in battery life comes at the cost of a decrease in computational complexity. Many such devices only contain 8- or 16-bit processors with very limited RAM memory, and are unable to implement complex video encoding algorithms.

Some attempts have been made to implement video on low-complexity devices using traditional encoding methods. The best-known example of this is motion JPEG [23] (MJPEG), where a video is encoded as a series of JPEG encoded images. While MJPEG has become popular in devices such as digital cameras and cellphones, it is clearly not an ideal solution. For example, MJPEG does not take advantage of temporal correlation in a video sequence. In addition, JPEG image encoding still requires the source node to capture and temporarily store an entire raw video frame and perform a DCT [24] transform on each block of the image. While far less complex than motion vector calculations, these are still not insignificant operations. To address this completely, an entirely new system is required that encodes video at very low computational complexity, yet still takes advantage of temporal correlation within the video stream. Below, we demonstrate that CS can be used to design such a system.

2.2 Channel Constraints

A major challenge in WMSNs is compensating for lossy channels. Because WSN nodes transmit at lower power than other (nonenergy aware) types of wireless nodes, restricting ourselves to the WSN device will result in a lower SNR channel, resulting in more bit errors and packet losses. To achieve our target of high quality received video, rather than attempting to reduce or eliminate bit errors, we need a low-complexity technique that will compensate for the inevitable errors in the received video packets. We will discuss two aspects of wireless transmission that complicate the transmission of video, namely bit errors and multipath fading.

Bit Errors: It is well known that predictively encoded video is very susceptible to bit errors. It is also well known that with variable-length coding (i.e., Huffman coding [25]) a single bit error can cause the loss of entire blocks of data. In data networks, bit errors are usually dealt with using some form or ARQ or FEC. Both of these methods generally have an all-or-nothing approach to error correction, in that a received packet is either entirely correct or is *discarded and must be retransmitted*. However, even though video is less tolerant to bit errors than images, it is much more tolerant of bit errors than data networks [26, 27]. While the quality does decrease sharply when the BER increases beyond some threshold, for low levels of BER there is *no measurable decrease in video quality*.

This leads to an obvious tradeoff between the quality of the received video and the techniques used to reduce the BER. As is shown in Sect. 7, there is little or no effect in the perceivable quality in the received video for BER rates of up to 10^{-4} for H.264 or for 10^{-3} for CVS. One advantage of CS-encoded images and video is that, because of independence between samples within an image, many more errors can be tolerated before significant quality degradation is noted in the received video.

Important for WMSN applications, this shows us that, while channel conditions are very poor when using WNS nodes, in many cases we can *simply ignore* these errors. When we look at a system implementation, this intuitively tells us that even with low quality trancievers transmitting at low power such as a WSN system, we can still achieve high quality video by (i) *not* using a packet-level error detection scheme such as CRC, and (ii) assuring that the BER is "low enough." Below, we demonstrate that a CS system can potentially accomplish this.

Fading: While bit errors alone can cause major problems for video transmission if not accounted for, time correlation of those errors (i.e., fading) can also cause video quality to decrease significantly. One of the major problems associated with a fading channel is that, with correlation, the bit errors will tend to be "grouped together" within a single packet, rather than distributed randomly throughout the entire transmission. This can cause problems when using FEC to correct errors. When bit errors are grouped together within a single packet, there may be too many errors in that one packet for the FEC code to correct, leading to the incorrect decoding (and subsequent loss) of that packet.

A discussion of all of the effects of fading on wireless video communication could be an entire chapter (or book) by itself, and is beyond the scope of this chapter.

We bring it up here to emphasize that, while time correlation of errors causes problems for traditional video streaming systems, it has virtually *no impact* on a system that streams CS-encoded video. If errored video samples could be dropped without hindering the decoding of the correctly received samples, the "error grouping" effect of a fading channel would have no negative impact on received video performance, without the need for interleaving video samples. As we will demonstrate, CS can accomplish this very easily.

2.3 Data Rate Constraints

Unlike other types of sensor network traffic, multimedia traffic, specifically video traffic, offers some severe challenges. Protocols developed for sensor networks (i.e., Zigbee [28] or Bluetooth [29]) are designed for reduced power consumption. Generally, this is accomplished using techniques to power down the radio when it is not in use and reducing the transmit power. However, both of these approaches will lead to a significant decrease in the maximum data rate. For multimedia applications, this will require much more aggressive compression techniques, which are more complex and therefore require more power.

While there exist standardized medium access control (MAC) protocols that are able to provide a high enough data rate to wirelessly transmit multimedia content (e.g., 802.11 [30], WiMAX [31]), and there exist standardized protocols that are able to reduce the power consumption at each node to acceptable levels, achieving both at the same time is much more difficult. The standard power saving method in sensor network MAC protocols is for nodes to enter a sleep mode when they are not transmitting or receiving data. However, these sleep cycles reduce the maximum data rate, which is unacceptable when multimedia traffic is being transmitted.

CS-based systems cannot solve this problem directly. Indeed, we will show that from a purely rate-distortion perspective, H.264 performs far better than any CS system. However, as we will show later, rate-distortion performance itself does not give the entire story in resource-constrained systems. It is important to examine what causes this limit in data rate. For all MAC protocols, the protocol specifications determine some modulation and channel coding combination that will achieve a given BER at the receiver. However, we have shown above that CS-based systems are able to tolerate a relatively high BER. This means that we could potentially reduce the channel coding rate, increase the number of bits per symbol in the modulation technique or both to result in a higher data rate at the cost of a higher (but still tolerable) BER.

2.4 Cost Constraints

Finally, to be feasible in a large scale, WMSN nodes should be as inexpensive as possible. While a cost analysis is beyond the scope of this chapter, we mention this simply because, while more expensive processors and bigger batteries may solve many of the challenges posed above, this is not a realistic solution for WMSNs [1, 3]. As we have been stressing throughout this work, we would like to keep the cost of the WMSN node to around the cost of a comparable scalar WSN node not taking the actual camera into account, around $50 USD.

3 Compressed Sensing Basics

In this section, we introduce the basic concepts of compressed sensing as applied to image compression. We consider an image signal represented through a vector $\mathbf{x} \in \mathbb{R}^N$, where N is the number of pixels in the image and each element of the vector x_i represents the i^{th} pixel in the raster scan of the image. We assume that there exists an invertible transform matrix $\boldsymbol{\Psi} \in \mathbb{R}^{N \times N}$ such that

$$\mathbf{x} = \boldsymbol{\Psi}\mathbf{s} \tag{1}$$

where $\mathbf{s}$ is a K-sparse vector, i.e., $\|\mathbf{s}\|_0 = K$ with $K < N$, and where $\|\cdot\|_p$ represents p-norm. This means that the image has a sparse representation in some transformed domain, e.g., wavelet [32]. The signal is measured by taking $M < N$ samples of the element vectors through a linear measurement operator $\boldsymbol{\Phi}$, defined by

$$\mathbf{y} = \boldsymbol{\Phi}\mathbf{x} = \boldsymbol{\Phi}\boldsymbol{\Psi}\mathbf{s} = \tilde{\boldsymbol{\Psi}}\mathbf{s}. \tag{2}$$

We would like to recover $\mathbf{x}$ from measurements in $\mathbf{y}$. However, since $M < N$ the system is underdetermined. Hence, given a solution $\mathbf{s^0}$ to (2), any vector $\mathbf{s}^*$ such that $\mathbf{s}^* = \mathbf{s}^0 + \mathbf{n}$, and $\mathbf{n} \in \mathcal{N}(\tilde{\boldsymbol{\Psi}})$ (where $\mathcal{N}(\tilde{\boldsymbol{\Psi}})$ represents the null space of $\tilde{\boldsymbol{\Psi}}$), is also a solution to (2). However, it was proven in [15] that if the measurement matrix $\boldsymbol{\Phi}$ is sufficiently incoherent with respect to the sparsifying matrix $\boldsymbol{\Psi}$, and K is smaller than a given threshold (i.e., the sparse representation $\mathbf{s}$ of the original signal $\mathbf{x}$ is "sparse enough"), then the original $\mathbf{s}$ can be recovered by solving the optimization problem

$$\begin{aligned} &\underset{\mathbf{s}}{\text{minimize}} \quad \|\mathbf{s}\|_0 \\ &\text{subject to} \quad \mathbf{y} = \tilde{\boldsymbol{\Psi}}\mathbf{s} \end{aligned} \tag{3}$$

which finds the sparsest solution that satisfies (2), i.e., the sparsest solution that "matches" the measurements in $\mathbf{y}$.

Unfortunately, finding the *sparsest* vector $\hat{\mathbf{s}}$ using (3) is in general NP-hard [33]. However, for matrices $\tilde{\boldsymbol{\Psi}}$ with sufficiently incoherent columns, whenever this

problem has a sufficiently sparse solution, the solution is unique, and it is equal to the solution of the following problem:

$$\begin{aligned} \underset{\mathbf{s}}{\text{minimize}} \quad & \|\mathbf{s}\|_1 \\ \text{subject to} \quad & \left\|\mathbf{y} - \tilde{\boldsymbol{\Psi}}\mathbf{s}\right\|_2^2 < \varepsilon \end{aligned} \tag{4}$$

where ε is a small tolerance.

Formally, any sampling matrix $\boldsymbol{\Phi}$ must satisfy the uniform uncertainty principle (UUP) [15, 34]. The UUP formally states that if enough samples are taken, such that

$$M \geq K log N, \tag{5}$$

then for any K-sparse vector $\mathbf{s}$, the energy of the measurements $\boldsymbol{\Phi}\mathbf{s}$ will be comparable to the energy of $\mathbf{s}$ itself:

$$\frac{1}{2}\frac{M}{N} \cdot \|\mathbf{s}\|_2^2 \leq \|\boldsymbol{\Phi}\mathbf{s}\|_2^2 \leq \frac{3}{2}\frac{M}{N} \cdot \|\mathbf{s}\|_2^2. \tag{6}$$

To intuitively see the association between UUP and sparse reconstruction [34], suppose that (6) holds for sets of size 2K. If our K sparse vector $\mathbf{y}$ is measured as $\mathbf{y} = \boldsymbol{\Phi}\mathbf{s}_0$, then there cannot be any other K-sparse or sparser vector $\hat{\mathbf{s}}' \neq \mathbf{s}_0$ that leads to the same measurements. If there were such a vector, then the difference $\mathbf{h} = \hat{\mathbf{s}}_0 - \hat{\mathbf{s}}'$ would be $2K$-sparse and have $\boldsymbol{\Phi}\mathbf{h} = 0$. However, this is not compatible with (6).

Note that (4) is a convex optimization problem [35]. The reconstruction complexity equals $O(M^2N^{3/2})$ if the problem is solved using interior point methods [36]. Although more efficient reconstruction techniques exist (for example, those mentioned in Sect. 9.1), we only discuss specific reconstruction algorithms when necessary to understand the specific imaging or video system. Otherwise, the discussions presented here are independent of the specific reconstruction algorithm.

4 Compressive Imaging

The CS-based video compression schemes are similar to many traditional video encoders that use a combination of image encoding schemes and motion compression schemes. Because of this, before discussing CS video, it is important to understand CS-based imaging and how CS-encoded images behave in a real system.

4.1 Compressive Imaging Background

Since most images can be represented in a sparse domain (i.e., wavelet or DCT), they can be sampled and compressed using (2) and recovered using (4). In this section

we will examine some of the properties of images that have been compressed using this CS system, and how these *properties* can help address the challenges described in Sect. 2.

Effects of Approximate Sparsity: In Sect. 3, we stated that any K-sparse signal sampled using (2) that satisfies (5) can be recovered using (4). However, wavelet (or DCT) transformed images are only *approximately* sparse. For example, Fig. 1 shows the DCT coefficients of the Lena image [37] sorted in increasing order. While the image is clearly compressible, none of the DCT coefficients are *exactly* 0.

When we use (4) to reconstruct Lena with $M < N$, the reconstruction process will force the smaller coefficients to be exactly 0 [16], which will cause distortion in the reconstructed image. We can see how this affects the quality of the reconstructed image by measuring the effect of this sparse approximation on DCT transformed images. The results of this test are shown in Fig. 2. This figure was created by calculating the DCT transform of the Lena image, forcing the smallest coefficients to zero and calculating the inverse transform of the result. As the number of coefficients forced to 0 increases, the quality of the reconstructed image decreases.

In practice, this means that, unlike the sparse case described above, "exact" recovery is not possible. Instead, as more samples are used in the reconstruction (i.e., as M approaches N), the reconstructed image quality increases. This is demonstrated in Fig. 3, which shows the mean of the received quality over all of the images in the USC SIPI database [37]. These tests were done using the wavelet transform as the sparsifying transform and reconstructed using the gradient projection for sparse reconstruction GPSR [38] algorithm. As M is increased and more samples are used in the image reconstruction, the SSIM of the image approaches 1. It is also worth while to note that, rather than a "minimum" number of samples required to

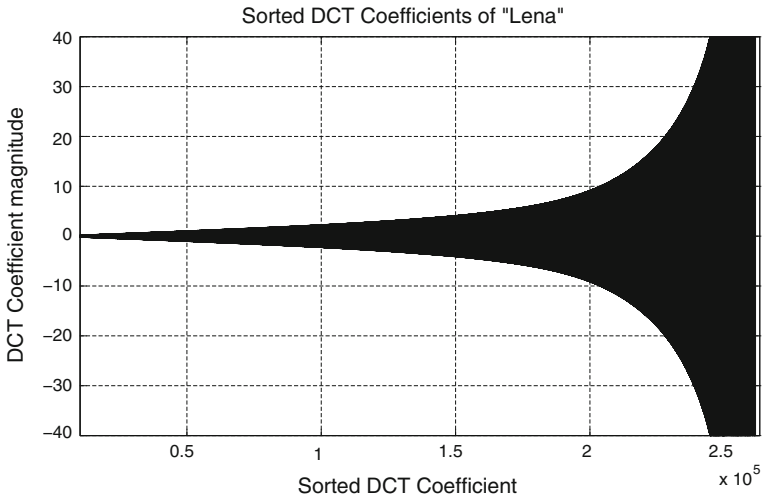


Fig. 1 DCT coefficients of Lena sorted in ascending order

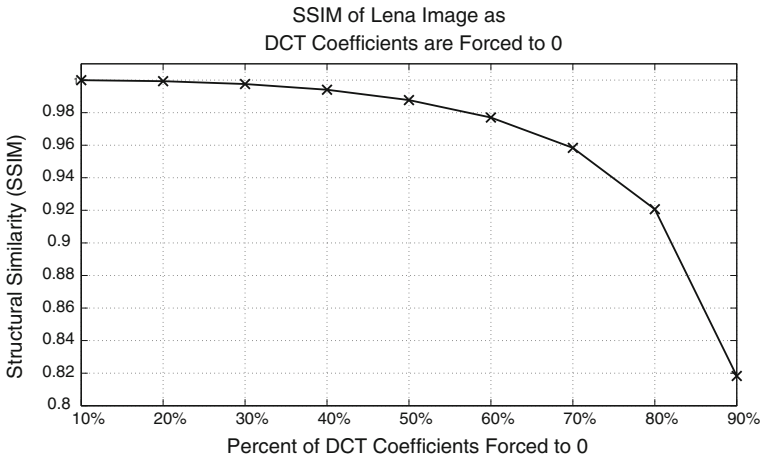


Fig. 2 SSIM of Lena after DCT transform, forcing the smallest coefficients to zero and inverse DCT transform

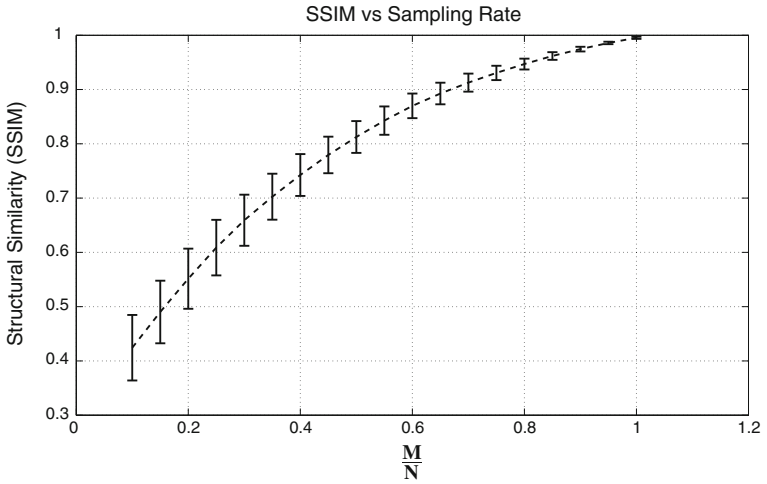


Fig. 3 SSIM Versus sampling rate $\frac{M}{N}$

correctly reconstruct the image, using fewer samples simply results in a lower quality reconstructed image.

Effects of Quantization: In general, CS theory assumes that the signal is compressed and recovered in the real domain. However, we are usually interested in transmitting a quantized version of the signal. Since the user chooses the value of M, which is arbitrary within a certain range, there is a tradeoff between transmitting *fewer samples encoded with more bits each* or transmitting *more samples encoded with fewer bits*. This is examined empirically (again over the images in the SIPI

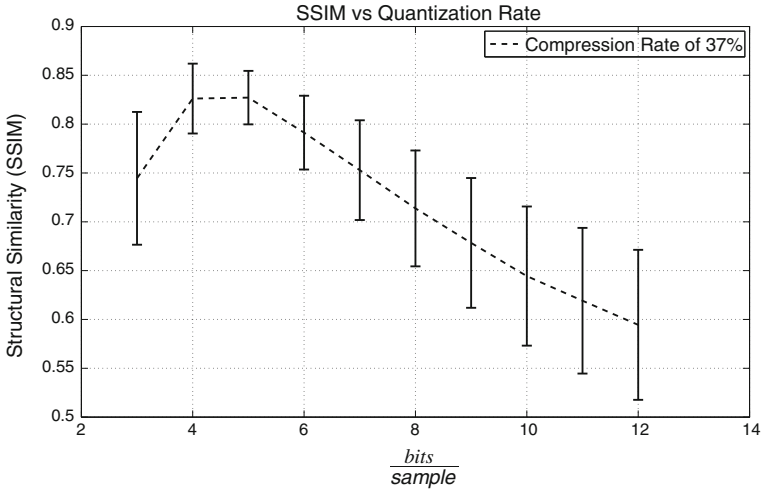


Fig. 4 SSIM Versus quantization bits

database), and is presented in Fig. 4. All of the original images are gray-scale images quantized at 8 bits per pixel. It is interesting to note that the highest quality reconstruction occurs when the number of samples per symbol is *lower* than the number of samples per pixel in the original image. This means that there is less precision in the samples than in the original pixels, yet we are still able to reconstruct the image with high quality.

This result is in agreement with [16], which shows that CS reconstruction is generally very resistant to low power noise, such as quantization noise. Suppose we have a set of measurement samples $\mathbf{y}^{\#} = \boldsymbol{\Phi}\mathbf{x} + \mathbf{n}$ corrupted by noise, where $\mathbf{n}$ is a deterministic noise term, and is bounded by $\|\mathbf{n}\|_2 < \varepsilon$. As long as $\boldsymbol{\Phi}$ obeys (6), then the value of $\mathbf{x}^{\#}$ reconstructed using (4) from $\mathbf{y}^{\#}$ will be within

$$\|\mathbf{x}^{\#} - \mathbf{x}\| \leq C \cdot \varepsilon, \tag{7}$$

where C is a "well behaved" constant.[1]

While the full proof of this is beyond the scope of this chapter, it is easy to see why $\boldsymbol{\Phi}\mathbf{x}^{\#}$ will be within 2ε of $\boldsymbol{\Phi}\mathbf{x}$ using the triangle inequality. Specifically,

$$\|\boldsymbol{\Phi}\mathbf{x}^{\#} - \boldsymbol{\Phi}\mathbf{x}\|_2 \leq \|\boldsymbol{\Phi}\mathbf{x}^{\#} - \mathbf{y}\|_2 + \|\boldsymbol{\Phi}\mathbf{x} - \mathbf{y}\|_2 \leq 2\varepsilon. \tag{8}$$

This can be seen in Fig. 5, which represents a system that samples a variable $\mathbf{x} \in \mathbb{R}^2$ with a sampling matrix $A \in \mathbb{R}^{1\times 2}$. The line represents $\mathbf{y} = \boldsymbol{\Phi}\mathbf{x}$, while the diamond represents the ℓ_1 norm ball. The two dashed lines represent the maximum variation in the samples when corrupted by additive noise of magnitude ε. The point where

[1] For practical systems, C is a small constant between 5 and 10 [16].

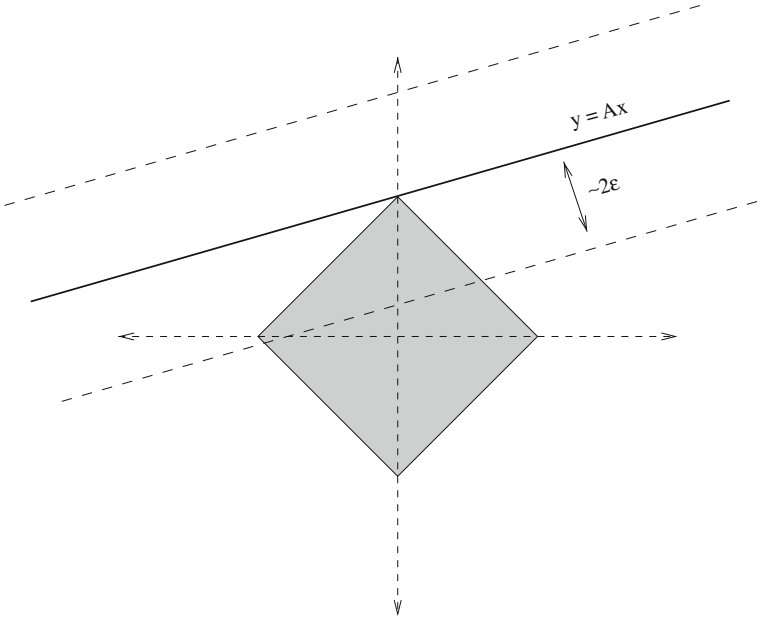


Fig. 5 Geometric interpretation of ℓ_1 norm minimization

the smallest norm ball intersects the line is the sparsest solution, and is therefore the solution to (4). While this is a simplistic example, it is easy to see that, at least in this case, the error in the reconstructed sample will result in a small variation in the magnitude of the reconstructed signal. In the system represented in Fig. 5, the magnitude of ε would have to be about $\frac{1}{3}$ of the signal power before an incorrect "corner" of the norm ball is selected.

Effects of Bit Errors: Though we accurately model the video distortion when there are no errors [27, 39], any bit errors may add further distortion to the received image. As shown in Fig. 6, the video does not have to be received perfectly for it to be acceptable at the receiver. Figure 6 shows the reconstruction quality of images transmitted through a binary symmetric channel. At low BER rates, there is almost no effect in the received SSIM. As the BER increases above some threshold, however, the video quality drops off significantly.

Sampling Complexity: Traditional image compression schemes generally partition an image into smaller components, and compress each of these components individually. An example of this can be seen in the popular JPEG standard. A JPEG encoder first divides an image into 8×8 pixel blocks. Then each of these 64 pixel groups are transformed using a DCT transform. JPEG2000 [40] is based on a 2D wavelet transform. However, the actual implementation of that 2D wavelet transform is based on a series of 1D wavelet transforms [41] of each column and row sequentially. In both of encoders, the processing is limited to a subset of the image pixels at any given time.

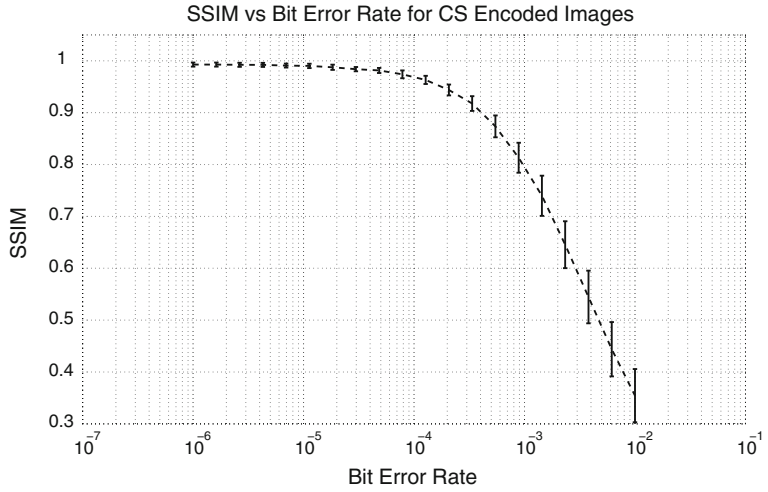


Fig. 6 Compressed sensed images reconstructed with bit errors

Methods of dividing imaging problems into subproblems are necessary because of the computational complexity required to encode realistic sized images with non-linear transform operations. Like JPEG and JPEG2000, a CS imaging system must manage this complexity as part of the development of any implementable system. For example, a direct implementation of (2) requires the creation and storage of $\boldsymbol{\Phi} \in \mathbb{R}^{M \times N}$. Assume we are dealing with a 512×512 pixel image, and that M is set at $\frac{N}{5}$ (representing 80 % compression). This will result in a $\boldsymbol{\Phi}$ matrix that is $52{,}429 \times 262{,}144$. A direct implementation would require matrix multiplication with a matrix of over 13 billion elements, which is clearly not practical.

This can be avoided by sampling using a scrambled block Hadamard matrix [42], defined as

$$Y = H_{32} \cdot X, \tag{9}$$

where Y represents image samples (measurements), H_{32} is the 32×32 Hadamard matrix and X the matrix of the image pixels. The matrix X has been randomly reordered and shaped into a $32 \times \frac{N}{32}$ matrix. Then, M samples are randomly chosen from Y and transmitted to the receiver. The receiver then uses the M samples along with the randomization patterns for both randomizing the pixels into $\mathbf{x}$ and choosing the samples out of Y (both of which can be decided before network setup). The result is a sampling system that is much lower complexity, yet is equivalent to the performance of (2).

4.2 Single Pixel Camera

A major step in making the theoretical CS imaging applications more practical was the development of the single pixel camera [43]. The single pixel camera is able to simultaneously measure and compress images in hardware with very low complexity. The camera uses a Texas Instruments digital mircomirror device (DMD) [44] to reflect an image onto a single photodiode. The DMD is able to individually change the angle of each mirror from a bank of 1027×768 mirrors to either $+12$ or -12° from horizontal. Using a biconvex lens, this allows the system to aim a subset of the mirrors at the photodiode. The output of this photodiode is then amplified and quantized to produce a single CS sample. This process is then repeated to produce all M samples. Each of these samples is then passed through an analog-to-digital converter, and either transmitted or stored in memory.

For WMSN applications, the major advantage of this system is the simplicity compared to other image sampling methods. The entire camera-encoder system consists of a single DMD and an analog-to- digital converter. All of the signal processing (i.e., the linear combinations of pixels) is done implicitly when the intensity at the photo-detector is measured. Another less obvious advantage is that, since only a single photodiode is used, a infrared imaging system can be built without increasing the cost significantly over the visual light system.

4.3 Wireless Transmission of CS-Encoded Images

There are two very important advantages that CS imaging has over JPEG imaging. First, CS imaging can compress an image with far lower computational complexity. While it is difficult to get an accurate quantitative measurement between the two, note that, neglecting the DMD, the entire CS imaging described in Sect. 4.2 is lower by a factor of $\frac{M}{N}$ than the complexity required *simply to capture the image in a JPEG-based device*.

CS-based encoders also perform comparatively well in a noisy channel. CS-encoded samples constitute a random, incoherent combination of the original image pixels. This means that, unlike traditional wireless imaging systems, no individual sample is more important for image reconstruction than any other sample. Instead, *the number of correctly received samples* is the main factor in determining the quality of the received image. This naturally leads to scheme where, rather than trying to correct bit errors, we can instead *detect* errors and simply drop samples that contain errors. This is demonstrated in Fig. 7, where a set of images [37] are encoded using CS and transmitted over a lossy channel. For the purpose of demonstration, we assume that there is a genie at the receiver that is able to perfectly detect when a sample is received incorrectly. We then show the image reconstruction quality with and without those samples. Clearly, based on Fig. 7, simply removing those samples

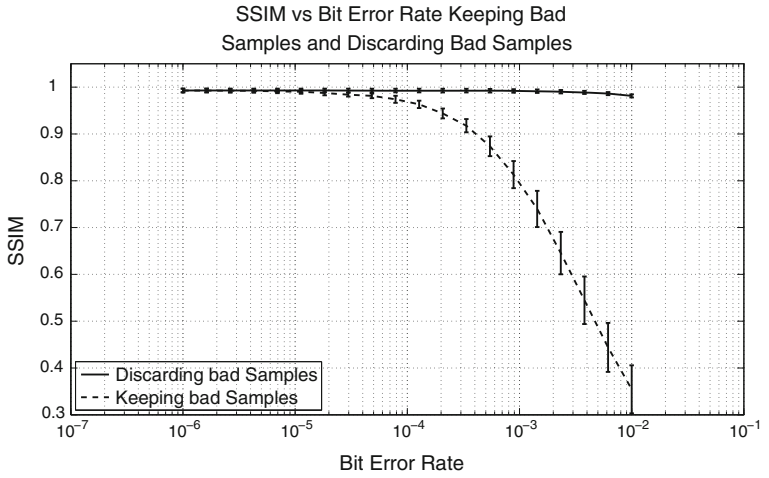


Fig. 7 Compressed sensed images reconstructed with and without incorrect samples

results in a far better reconstruction quality that if those incorrect samples are used in the reconstruction process.

While it is easier to deal with errors in a CS system, the errors that are used in the reconstruction process do not have as much impact on the reconstructed image quality as when using a JPEG system. A small amount of random channel errors does not affect the perceptual quality of the received image *at all*, since, for moderate bit error rates, the greater sparsity of the "correct" image will offset the error caused by the incorrect bit. This is demonstrated in Fig. 7. For any BER lower than 10^{-4}, there is *no noticeable drop in the image quality*. For BER levels of 10^{-3} or lower, the SSIM is above 0.8, which is an indicator of good image quality. If the BER is kept below 10^{-5}, there is virtually no distortion in the received image.

This has important consequences and provides a strong motivation for studying compressive wireless video streaming in WMSNs. This inherent resiliency of compressed sensing to random channel bit errors is even more noticeable when compared directly to JPEG. Figure 8 shows the average SSIM of the SIPI images [37] transmitted through a binary symmetric channel with varying BER. These values were calculated by encoding a set of images using both encoders, transmitting those files through a simulated channel, and reconstructing the image including the bit errors.

The quality of CS-encoded images degrades gracefully as the BER increases, and is still good for BERs as high as 10^{-3}. Instead, JPEG-encoded images very quickly deteriorate. This is visually emphasized in Fig. 9, which shows an image taken at the University at Buffalo encoded with CS (above) and JPEG (below) and transmitted with bit error rates of 10^{-5}, 10^{-4}, and 10^{-3}. The difference is stunning—the effect of channel errors is disruptive for structured data like JPEG-encoded images. The reader will easily realize that the effect of channel errors on predictively encoded video is even more disruptive, since even low bit error rates can lead to the loss of

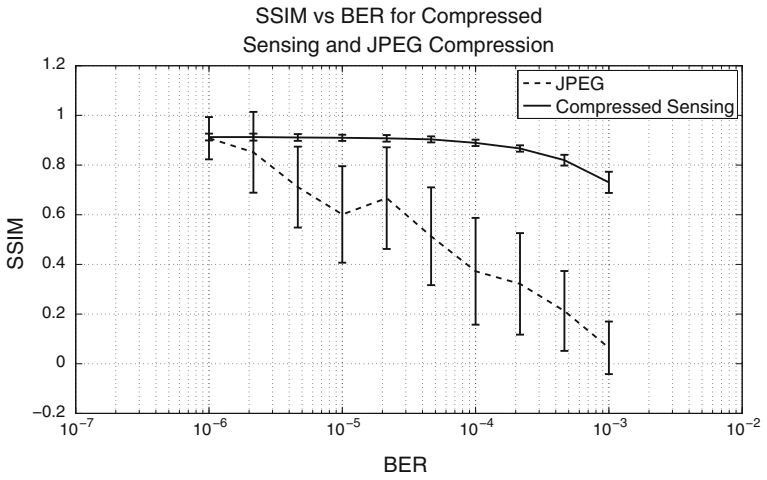


Fig. 8 Structural similarity (SSIM) Versus bit error eate (BER) for compressed sensed images, and images compressed using JPEG

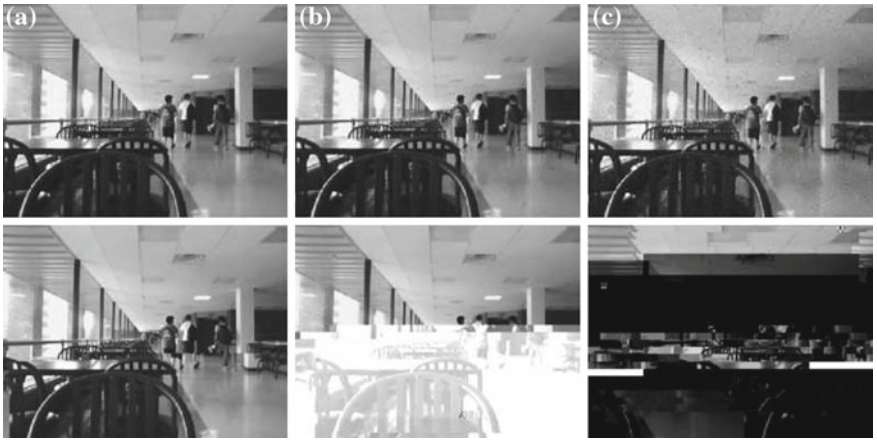


Fig. 9 Image compressed with CS (*above*) and JPEG (*below*) for BER. **a** 10^{-5}; **b** 10^{-4}; **c** 10^{-3}

I frames, causing the decoder to be unable to decode long sequences of frames that depend on the *I* frame.

5 Video Encoding with Compressed Sensing

We saw in the previous section that CS-encoded images are more resilient to channel errors than JPEG encoded images. Based on this, the most straightforward video encoding scheme is to take each frame of a video individually, treat it as an image

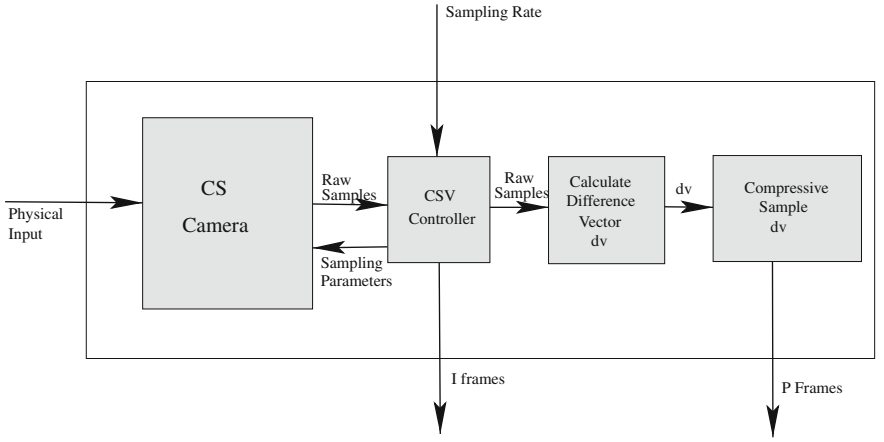


Fig. 10 Block diagram for CS video decoder

and use CS image encoding schemes. Although this approach seems very simplistic, it is conceptually analogous to the very common MJPEG.

While such a system would indeed have low complexity at the sensor nodes, and the error resilience of CS-encoded images is much higher, (taking care of most of our challenges), such a scheme ignores the temporal correlation between consecutive frames. We know that because of complexity, motion vectors used in traditional schemes are not appropriate for these applications. There are however methods for taking advantage of this correlation without using motion vectors. Below, we describe three computationally "easy" techniques based on state-of-the-art CS video encoders.

5.1 Exploiting Temporal Correlation via Difference Vectors

To take advantage of temporal correlation, we consider the algebraic difference between the CS samples, as in [39]. The motivation behind this is that a CS-encoded image is simply a series of linear combinations of subsets of the pixels of an image, which is represented by the multiplication by the sampling matrix $\boldsymbol{\Phi}$. Now assume that we have two frames $\mathbf{x_i}$ and $\mathbf{x_{i+1}}$. For most CS applications, it is assumed that the transmitting sensor node does not have access to the raw image data; in this case $\mathbf{x_i}$ and $\mathbf{x_{i+1}}$. Instead, the sensor node only has access to $\mathbf{y_i} = \boldsymbol{\Phi}\mathbf{x_i}$ and $\mathbf{y_{i+1}} = \boldsymbol{\Phi}\mathbf{x_{i+1}}$. However, as long as $\boldsymbol{\Phi}$ is kept constant, it is easy to see that if we calculate a difference vector $\mathbf{dv}$ as

$$\mathbf{dv_{i+1}} = \mathbf{y_{i+1}} - \mathbf{y_i}, \tag{10}$$

that this is equivalent to

$$\begin{aligned} \mathbf{dv_{i+1}} &= \boldsymbol{\Phi}\mathbf{x_{i+1}} - \boldsymbol{\Phi}\mathbf{x_i} \\ &= \boldsymbol{\Phi}(\mathbf{x_{i+1}} - \mathbf{x_i}), \end{aligned} \tag{11}$$

which is the same as if we had sampled the difference between the two frames explicitly.

Then, each dv_{i+1} is *again compressively sampled* and transmitted. If the image being encoded x_{i+1} and the reference image x_i are very similar (i.e., have a very high correlation coefficient), then dv_{i+1} will be sparse (in the domain of compressed samples) and have less variance than either of the original images. The main compression of the difference frames comes from the above properties and is exploited in two ways. First, because of the sparsity in the difference frame, it can be further compressed using CS. The number of samples needed is based on the sparsity as in the CS sampling of the initial frame. Second, the lower variance allows us to use fewer quantization levels to accurately represent the information, and therefore fewer bits per sample. A block diagram representing this encoding scheme is shown in Fig. 10.

Formally, **dv** is compressed using (2), quantized and transmitted. The number of samples m needed to represent **dv** after it is compressed is proportional to the *sparsity* K of **dv** and defined as $m \approx K\ log(N)$ where N is the length of **dv**. For videos with very high temporal correlation such as security videos, the **dv** will also have very low variance, allowing for a lower quantization rate Q. In the simulations reported in this paper, we used $Q = 3$.

In terms of compression ratio, the effectiveness of this scheme depends on the temporal correlation between frames of the video. The compression of each of these schemes (at the same received video quality) was compared to basic CS compression (i.e., using I frames only) for three videos. The videos chosen were Foreman (representing high motion) and two security videos; one monitoring a walkway with moderate traffic (moderate motion) and one monitoring a walkway with only light traffic (low motion), and the percentage improvement, calculated as $\frac{\text{Size without P frames}}{\text{Size with P frames}} \times 100$ is presented in Table 1. While the compression of the high motion video can be increased by 172 %, the moderate and low motion security videos (which represent typical application scenarios for our encoder) show far more improvement by using the P frames.

5.1.1 Video Decoding

While the majority of the computational complexity of the system lies in decoding the video, a description of this system is relatively straightforward. A block dia-

Table 1 Compression gain using P frames

Amount of motion	low (%)	medium (%)	high (%)
Gain	556	455	172

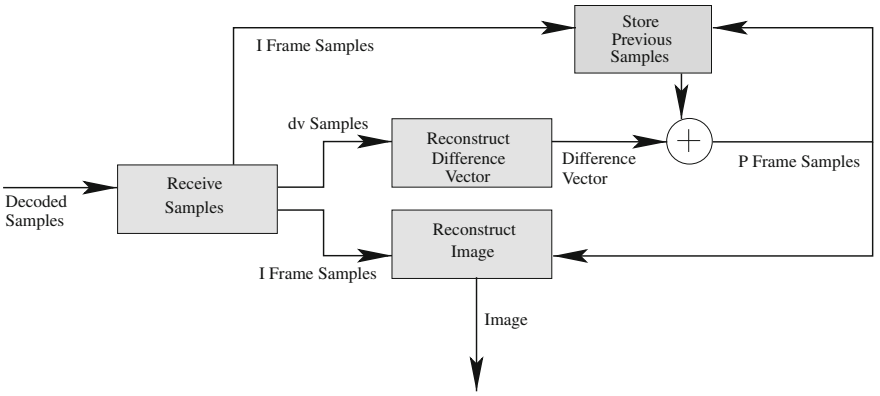


Fig. 11 Block diagram for CS video decoder

gram representing the CVS video decoder is shown in Fig. 11. If the received frame is an I frame, then the decoder simply solves (4) based on the received samples. These samples are also stored for use in decoding the P frames, and are defined as $\hat{\mathbf{y}}_\mathbf{I}$. If the received frame is a P frame, then the received samples represent a difference vector $\hat{\mathbf{dv}}$. Once $\hat{\mathbf{dv}}$ is reconstructed, again by solving (4), the samples of the P frame are calculated from $\hat{\mathbf{y}}_\mathbf{P} = \hat{\mathbf{dv}} + \hat{\mathbf{y}}_\mathbf{I}$, and the P frame can be reconstructed. Since the original $\mathbf{dv}$ is calculated as the difference between the samples of the P frame and I frame, the vector $\hat{\mathbf{y}}_\mathbf{P}$ should be very close to the original samples of the P frame, and can then be used to reconstruct the P frame.

Solving this system is based on some implementation of a convex solver. While the details of such a solver are beyond the scope of this chapter, we would like to note that common commercial solvers such as CVX [45] or SeDuMi [46] are not appropriate for a problem of this image reconstruction, as the dimension of the problem is too high. There do however exist solvers designed specifically for high dimension CS reconstruction, such as (GPSR) [38] or stagewise orthogonal matching pursuit (StOMP) [47]. For the decoders here, including the CVS video decoder, the specific implementation of the solver is not important.

6 Image Encoding and Recovery Using Compressed Sensing with CSEC and AP

We have seen that CS-encoded signals are resilient to channel errors. In this section, we examine a system that uses CS to actively protect the encoded images and video from channel errors. There are two main goals to this system.

- **Maintain Target Image Quality.** The CSEC portion of the system is charged with maintaining the image quality given a lossy channel. This system takes as

input both the number of packets expected to be lost due to collision or transmitter errors and the number of samples expected to be lost due to bit errors that would be detected by the AP system. Oversampling is then used to make up for these errors and allow the receiver to recover the image as if the original number of samples were sent. For example, assume that the transmitter intended to transmit 10,000 samples to the receiver to recover some image. Also assume that 5 % of the packets will be lost due to collision or transmission errors, and 3 % of the remaining samples will be lost due to bit errors, which results in a total loss rate of 7.85 %. By oversampling the signal to compensate for the expected loss, the total number of samples K can be found to be 10,852. This tells the transmitter that, based on the loss estimate of 7.85 %, if 10,852 samples are transmitted, roughly 10,000 samples will eventually be received correctly at the receiver. The details of the CSEC oversampling rate will be explained in detail in Sect. 6.1.

- **Minimize the Number of Transmitted Samples for a Target Desired Quality.** The AP portion of this system uses the *estimated* bit error rate of the channel to determine the optimal number of samples to include for each parity bit. This system will then use this information to determine the expected number of correctly received samples. This is done by analytically determining the optimal number of parity bits needed to maximize the number of correctly received samples at the receiver. The details of the AP calculation will be explained in detail in Sect. 6.2.

The basis for both of these systems is that the compressed samples which are created using the CS paradigm are all *equally important* and losing a single sample does not affect the receivers ability to be able to recover any other sample. Also, the specific samples chosen for use in the recovery of the image are arbitrary. This means that, if a sample is lost, a different sample can be transmitted in its place with no effect on the quality of the recovered image.

6.1 Erasure Channel Coding Using Compressed Sensing

CSEC has the ability to recreate the signal with some degradation even if the errors exceed the threshold for recovery. This is possible by oversampling the signal to compensate for the losses. The total number of samples needed, K, depends on the channel loss probability and is given by

$$K = \frac{m}{(1-p)}, \tag{12}$$

where K is the number of samples needed for a lossless transmission and is a function of the sparsity of the signal and m is the number of correctly received samples needed to achieve a desired image quality. Basically, the coding is done such that the number of correctly received samples for a given error probability p is equal to the number of samples in the original signal without errors, i.e., $(1-p)\cdot(K) = m$.

To see how the recreation of an image is affected by oversampling, we simulated the recovery of a 32×32 image under three conditions; no loss, 20 % sample loss, and CSEC with 20 % oversampling. The sampling matrix is assumed to be Gaussian with mean zero and variance $\frac{1}{1024}$. An image size of 32×32 was chosen. The number of measurements in lossless case (m) is taken to be 800. We choose PSNR as the reconstructed image quality indicator, which is defined as

$$PSNR = 10 \cdot \log_{10}\left(\frac{MAX_I^2}{MSE}\right), \tag{13}$$

where MAX_I is the maximum possible pixel value for each frame. MSE is the mean squared error, which is defined as

$$MSE = \frac{1}{mn}\sum_{i=1}^{m-1}\sum_{j=1}^{n-1}\|I(i,j) - K(i,j)\|^2. \tag{14}$$

We use the Discrete Cosine Transform (DCT) as the sparsifying transform and CVX to solve the reconstruction problem (4).

In the lossless case, the PSNR is found to be 21.40 dB. With a sample loss rate of 20 % and no oversampling, the PSNR drops to 16.78 dB. Finally, with 20 % loss and 20 % oversampling, the PSNR value is 20.10 dB. Comparing PSNR values of the lossless and oversampled recovery cases, we can see that the images in both cases have similar reconstruction quality. The differences between the errorless case and the oversampling case can be accounted for by variations in the sampling matrix, which was different for each image.

6.2 Adaptive Parity-based Transmission

It was shown for images [26] that in CS, the transmitted samples constitute a random, incoherent combination of the original image pixels. This means that, unlike traditional wireless imaging systems, no individual sample is more important for image reconstruction than any other sample. Instead, the number of correctly received samples is the only main factor in determining the quality of the received image. Because of this, a sample containing an error can simply be discarded and the impact on the video quality is negligible as long as the amount or errors is small. This can be realized by using even parity on a predefined number of samples, which are all dropped at the receiver or at an intermediate node if the parity check fails. This is particularly beneficial in situations when receiver or at an intermediate node if the parity check fails. This is particularly beneficial in situations when the BER is still low, but too high to just ignore errors. To determine the amount of samples to be jointly encoded, the amount of correctly received samples is modeled as

$$C = \left(\frac{Q \cdot b}{Q \cdot b + 1}\right)(1 - BER)^{Q \cdot b}, \tag{15}$$

where C is the estimated amount of correctly received samples, b is the number of jointly encoded samples, and Q is the quantization rate per sample. To determine the optimal value of b for a given BER, (15) can be differentiated, set equal to zero and solved for b, resulting in

$$b = \frac{-1 + \sqrt{1 - \frac{4}{log(1 - BER)}}}{2Q}. \tag{16}$$

The optimal channel encoding rate can then be found from the measured/estimated value for the end-to-end BER and used to encode the samples based on (15). The received video quality using the parity scheme described was compared to different levels of channel protection using rate compatible punctured codes (RCPC). Specifically, we use the $\frac{1}{3}$ mother codes discussed in [12]. Briefly, a $\frac{1}{3}$ convolutional code is punctured to decrease the amount of redundancy needed for the encoding process. These codes are punctured progressively so that every higher rate code is a subset of the lower rate codes. For example, any bits that are punctured in the $\frac{2}{5}$ code must also be punctured in the $\frac{1}{2}$ code, the $\frac{2}{3}$ code, and so on down to the highest rate code ($\frac{8}{9}$, in this case). Because of this setup, the receiver can decode the entire family of codes with the same decoder. This allows the transmitter to choose the most suitable code for the given data. Clearly, as these codes are punctured to reduce the redundancy, the effectiveness of the codes decreases as far as the ability to correct bit errors. Therefore, we are trading BER for transmission rate.

Figure 12 shows the adaptive parity scheme compared to RCPC codes. This figure shows received video quality of a video encoded using CS and then protected from errors using either the adaptive parity scheme or traditional convolutional codes.

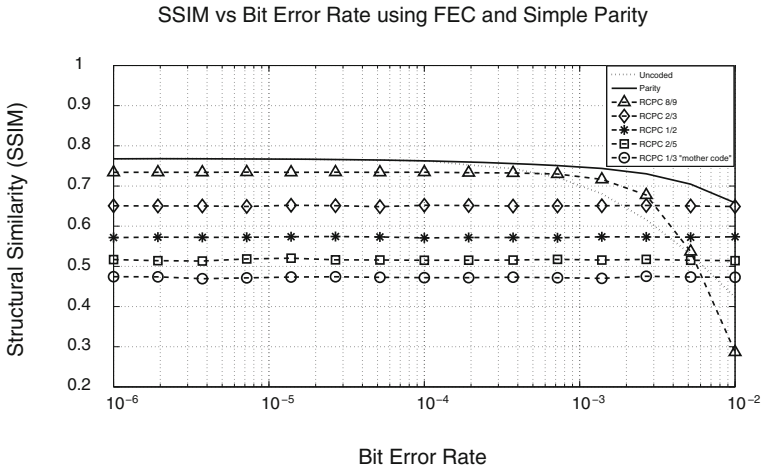


Fig. 12 Adaptive parity Versus RCPC encoding for variable bit error rates

Specifically, we use the $\frac{1}{3}$ mother codes described above and vary the puncturing rate to achieve all of the shown code rates. For comparison, the total number of transmitted bits is kept constant, so that as the strength of the FEC code increases, the number of CS samples is decreased to keep the number of bits constant. For all reasonable bit error rates, the adaptive parity scheme outperforms all levels of RCPC codes. The parity scheme is also much simpler to implement than more powerful forward error correction (FEC) schemes. This is because, even though the FEC schemes show stronger error correction capabilities, the additional overhead does not make up for the video quality increase compared to just dropping the samples which have errors.

7 Energy-Rate-Distortion Analysis of CVS

A question that naturally arises is how is the performance of CVS compares to that of H.264. The rate-distortion performance of CVS is not comparable to state-of-the-art encoders such as H.264. However, traditional rate-distortion analysis is inadequate to account for the computational and power limitations of mobile/sensing devices—the better rate-distortion performance of H.264 comes at a high *power* and *computational cost*. In addition, the CVS encoder is inherently resilient to channel errors, while the highly structured video representation of H.264 makes it highly vulnerable in lossy channels. Therefore, unlike H.264, CVS does not require strong channel coding and the resulting overhead. Different from previous work on low-complexity encoding, we jointly consider the effects of (i) processing on resource-constrained devices and of (ii) wireless transmission on the performance of wireless encoders, and conduct an experiment-driven analysis of the rate-power-distortion performance of different streaming systems designed for embedded wirelessly networked devices.

Intuitively, for a fixed energy budget, as more energy is allocated to the encoder (resulting in less compression and a video of better quality), less energy is available to transmit that video over a wireless link, which would potentially result in an increased bit error rate and lower quality at the receiver. Conversely, as more energy is allocated to transmission, less energy is available to encode the video, resulting in a lower quality video.

To analyze the rate-distortion performance of video encoders, we must first develop a model that accurately predicts the effect of compression and bit errors on the video quality. In a lossless channel, video distortion can be modeled [27] as

$$\alpha(r_v) = D_0 - \frac{\Theta}{r_v - R_0}, \tag{17}$$

where D_0, Θ and R_0 are video dependent constants determined through linear least squares estimation techniques.

Though this model works very well when there are no errors, any bit errors can decrease the quality of the received video. Unlike typical data networks, however,

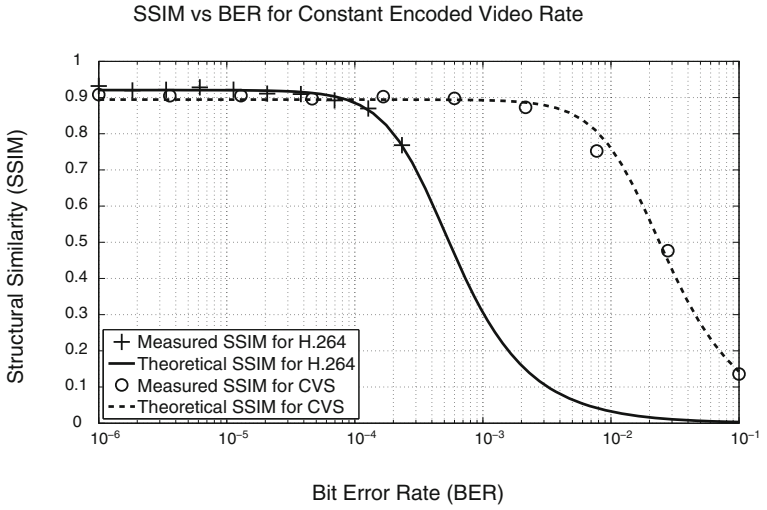


Fig. 13 SSIM Versus BER for H.264 and CVS encoders

the video does *not* have to be received perfectly, i.e., a moderate level of distortion is acceptable. This can be seen by observing a plot of the received video quality as a function of the bit error rate of the received video, as shown in Fig. 13. For this plot, the videos were encoded at a quality of 0.9 as measured in structural similarity (SSIM)[2] [48], transmitted through a binary symmetric channel with varying bit error rates (BER) and then decoded. For low BER, there is almost no effect in the received SSIM. After the BER increases past a certain level, however, the video quality drops off significantly.

Based on this observation, we have modeled the error performance as a low-pass filter using

$$U(r_{ch}, r_v) = \frac{\alpha(r_v)}{\sqrt{1 + \tau^2 (BER(r_{ch}, r_v))^2}} \tag{18}$$

where r_{ch} is the channel coding rate (in $\frac{bits\ in}{bits\ out}$), r_v is the encoded video rate in kbit/s, $U(r_{ch}, r_v)$ is the quality of the received video in SSIM as a function of r_{ch} and r_v. The encoder-dependent constant τ is used to indicate where the quality begins to decrease.

As is clear from Fig. 13, using (18) to represent the video quality allows us to very clearly compare the error resilience of different video encoders by observing the cutoff point (modeled by the variable τ) of different encoders. Clearly, CVS can tolerate a far higher BER before there is any noticeable effect on the reconstructed

[2] The SSIM index [48] is preferred to the more widespread peak signal to noise ratio (PSNR), which has been recently shown to be inconsistent with human eye perception [48, 49].

quality than H.264. As we will see below, this increase in acceptable BER translates to a decrease in the required transmission energy for the same reconstructed quality.

7.1 SNR Model

Consider the energy budget per frame E_B as the energy available to the system during each frame period $t_f = \frac{1}{fps}$ where fps represents the number of frames per second of the video. We can then express the average energy available for frame transmission as

$$E_E(r_v) = E_{E,max} \cdot t_e(r_v), \tag{19}$$

where $E_{E,max}$ is the maximum energy available to the encoder during the frame period, and $t_e(r_v)$ is the processor load, i.e., the time fraction of a frame that the encoder needs to encode video at rate r_v. The transmitted energy per video frame E_T is defined as

$$E_T = (E_B - E_E(r_v)), \tag{20}$$

i.e., the total energy available reduced by the energy needed to encode the video.

For the encoders considered in this paper, the empirical models

$$t_e(r_v) = a\, r_v + b, \tag{21}$$

and

$$t_e(r_v) = c\ - \frac{T}{r_v + d}, \tag{22}$$

accurately model the processor load as a function of the encoded video rate. The time $t_f = \frac{1}{fps}$, defined as the inverse of the framerate of the video, is used as the maximum allowed encoding time, i.e., the mean encoding time per frame for a real-time video must be less than t_f. The actual encoding time per frame, t_v is measured or estimated and compared to t_f. We can then find the value $t_e = \frac{t_v}{t_f}$ which represents the fraction of time used to encode each frame.

Based on this, we can then give the SNR model as

$$SNR(r_{ch}, r_v) = \frac{L \cdot r_{ch} \cdot d_{free} \cdot (E_B - E_E(r_v))}{N_0 \frac{r_v}{r_{ch} \cdot fps}}, \tag{23}$$

where L is the path loss, N_0 is the noise power and d_{free} is the free distance of the channel code r_{ch}. As r_v increases, the energy needed to encode the video increases while the transmission energy per bit decreases, causing the SNR to decrease.

7.2 Energy-Rate-Distortion Comparison

In Figs. 14 and 15, we show the energy-rate-distortion comparison between H.264 and CVS. We consider both the case where the video is originating at a relatively high powered system with $E_{E,max} = 0.5J$ (i.e., the energy to encode a frame on a desktop or laptop computer) in Fig. 14, and with $E_{E,max} = 0.167mJ$ (i.e., the energy to encode a frame on a smart phone) in Fig. 15.

These figures show two important comparisons between the two encoders. Traditional rate-distortion evaluation is equivalent to evaluating both of the encoders at the maximum power budget (the point at the far right of the graph). If this were all that mattered, then clearly H.264 would "outperform" CVS. However, as the energy budget is decreased, we see that CVS is still able to encode and transmit very high quality video, even when the energy budget is too low for H.264 to encode any video at all. In any system where using less energy overall is preferable, CVS far outperforms H.264.

8 Rate Control of CVS Encoded Video

To truly achieve a networked video system, we must define how a set of video sources can distributively determine the optimal video encoding rate (i.e, the size of $\boldsymbol{\Phi}$ in (2)) so as to achieve the maximum sum video quality at the receiver. In this section, we introduce such a congestion-avoiding rate control mechanism for use with the compressed sensed video encoder (CSV). The rate control subsystem

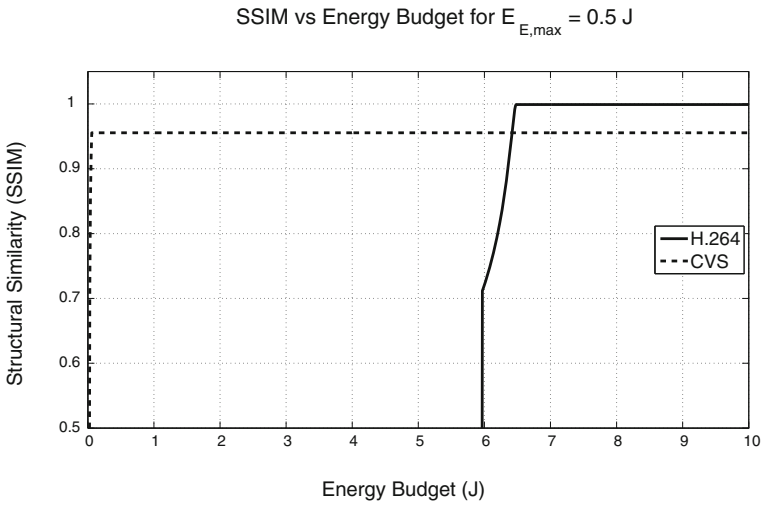


Fig. 14 SSIM Versus total energy budget for $E_{E,max} = 0.5\ J$

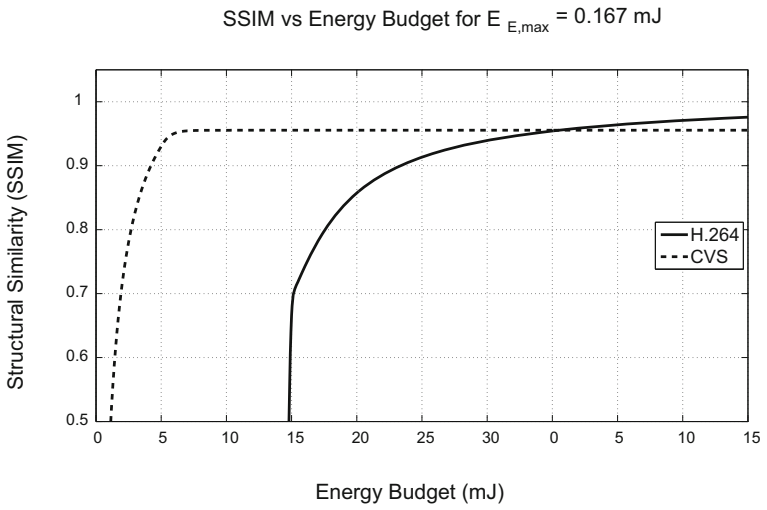


Fig. 15 SSIM Versus total energy budget for $E_{E,max} = 0.167\,mJ$

both provides fairness in terms of video quality and maximizes the overall video quality of multiple videos transported through the network. We then prove that this rate controller is indeed optimal in the domain of video quality.

8.1 CVS Rate Control Law

To avoid network congestion, a sending node needs to take two main factors into account. First, the sender needs to regulate its rate in such a way as to allow any competing transmission at least as much bandwidth as it needs to attain a comparable video quality as itself. Note that this is different from current Internet practice, in which the emphasis is on achieving fairness in terms of data rate (not video quality). Second, the sender needs to regulate its rate to make sure that packet losses due to buffer overflows are reduced, which can be done by reducing the overall data rate if it increases to a level that the network can not sustain.

To determine congestion, the round trip time RTT is measured for the transmitted video packets, where RTT is defined as the amount of time it takes for a packet to go from the source to the destination and a small reply packet to go from the destination back to the source. The change in RTT is measured as

$$\widetilde{\Delta RTT_t} = \frac{\sum_{i=0}^{N-1} a_i \cdot RTT_{t-i}}{N \cdot \sum_{i=0}^{N-1} a_i} - \frac{\sum_{i=1}^{N} a_i \cdot RTT_{t-i}}{N \cdot \sum_{i=1}^{N} a_i}, \tag{24}$$

which represents the difference of the weighted average over the previous N received RTT measurements with and without the most recent measurement. The weights a_i are used to low-pass filter the round trip time measurements, to give more importance to the most recent RTT measurements and to make sure that the protocol reacts quickly to current network events, while averaging assures that nodes do not react too quickly to a single high or low measurement.

The CVS video encoder generates two types of video frames; the I frame, which is an intra-encoded frame, and the P frame, which is an inter-encoded frame. The I frames are *independently encoded*, i.e., they are encoded using only the data contained within a single frame allowing these frames to be decoded independently of the previous frame. However, I frames do not take advantage of correlation between frames resulting in lower rate-distortion performance. P frames on the other hand are encoded based on previous frames by leveraging the temporal correlation between frames. Although this results in smaller frame sizes, it also allows errors to propagate from one frame to the next [27].

We present a novel approach in which the sampling rate γ_I of the video is used to control the data rate. Since γ_I is linearly proportional to the compression of the I frames (as seen in Fig. 16), controlling γ_I controls the compression rate of the entire video and therefore the data rate of the video transmission. Because of this linear relationship, we can control the compression of the entire video by varying only the I frame video quality.

We model the quality of the received video stream with a three-parameter model [27]

$$D_I = D_0 + \frac{\theta}{\gamma_I - R_0}, \tag{25}$$

where D_I represents the distortion of the video. The parameters D_0, θ and R_0 depend on the video characteristics and quantization level Q and can be estimated from empirical rate-distortion curves via a linear least-square curve fitting.

The rate control is based on the marginal distortion factor δ, which is defined by

$$\delta = \frac{\theta}{(\gamma_I - R_0)^2}, \tag{26}$$

i.e., the derivative of (25) with respect to γ_I.

At the source node of each video transmission, the amount of data generated by the video source for the $(t+1)^{th}$ group of pictures is controlled through

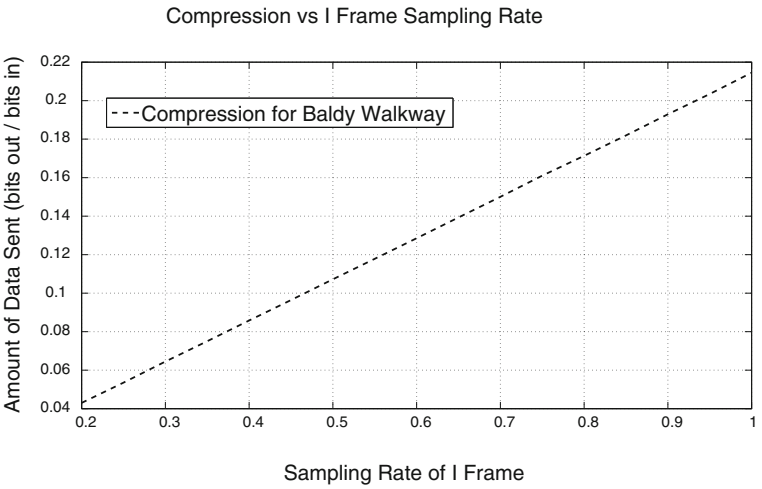


Fig. 16 Ratio of encoder output to encoder input Versus I frame sampling rate

$$\gamma_{I,t+1} = \begin{cases} \gamma_{I,t} - (1-\delta) \cdot \beta \cdot \widetilde{\Delta RTT}_t & \text{if } \widetilde{\Delta RTT}_t > \alpha \\ \gamma_{I,t} - \delta \cdot \kappa \cdot \widetilde{\Delta RTT}_t & \text{if } \widetilde{\Delta RTT}_t < -\alpha \\ \gamma_{I,t} & \text{else,} \end{cases} \tag{27}$$

where $\beta > 0$ and $\kappa > 0$ are both constants used to scale δ to the range of the sampling rate. α is a constant used to prevent the rate from oscillating with very minor changes in $\widetilde{\Delta RTT}_t$. The marginal distortion factor is used in (27) to promote fairness in terms of distortion. If there are two nodes transmitting video and both observe the same negative the sending node with the lower current video quality will take advantage of the decreased network congestion faster than the node that is transmitting at a higher rate by increasing its sampling rate more aggressively. The sampling rate more aggressively. The inverse is true for positive values of $\widetilde{\Delta RTT}_t$. This can be seen in Fig. 16. At lower compression levels, a change in the rate has a higher impact on the received image quality than an equal change will have at a higher rate. Similarly, $1 - \delta$, results in a function which has low values at low rates, and higher values at higher rates. The term $1 - \delta$ is then used to prevent a node from decreasing the rate significantly when the rate is already low, but encourage the node to decrease the rate when the data rate is already high.

8.2 The Optimality of the C-DMRC Rate Controller

Next, we analyze the performance of the rate controller presented in Sect. 8. We represent the network as a set $\mathcal{N}$ of nodes. The set $\mathcal{L}$ represents the set of all links in the network.

$$\begin{aligned} &\underset{\boldsymbol{\gamma_I}}{\text{maximize}} && \sum_{s\in\mathcal{S}} U_s(\gamma_{I,s}) \\ &\text{subject to} && \sum_{i:l\in\mathcal{L}(i)} \tau_i \gamma_{I,i} \le c_l, \quad \forall l \in \mathcal{L} \end{aligned} \tag{28}$$

where $\boldsymbol{\gamma_I} = [\gamma_{I,1}, \gamma_{I,2}, \ldots, \gamma_{I,|\mathcal{S}|}]$ is the vector of sampling rates for all sources, τ_i is a constant that maps sampling rates to data rates, i.e., $x_i = \tau_i \gamma_{I,i}$, $U_i(\gamma_{I,i}) = D_{0,i} + \frac{\theta_i}{\gamma_{I,i} - R_{0,i}}$ is the quality of video source i at sampling rate $\gamma_{I,i}$ and c_l represents the capacity of link l. Since $U_i(\gamma_{I,i})$ is a concave function and the constraints are affine in the rate variables, the problem is convex. We modeled the problem (28) with CVX [45] and solved it as a semidefinite program using SeDuMi [46].

We can now prove the following lemma.

Lemma 1 The rate control equation update (27) converges to a distributed solution to (28).

Proof The Lagrangian of (28) is defined as [35]

$$\begin{aligned} L(\boldsymbol{\gamma_I}, \lambda) &= \sum_{s\in\mathcal{S}} U_s(\gamma_{I,s}) - \sum_{l\in\mathcal{L}} \lambda_l \left(\sum_{s\in\mathcal{S}(l)} \tau_s \gamma_{I,s} - c_l \right) \\ &= \sum_{s\in\mathcal{S}} \left(U_s(\gamma_{I,s}) - \tau_s \gamma_{I,s} \sum_{l\in\mathcal{L}(s)} \lambda_l \right) + \sum_{l\in\mathcal{L}} \lambda_l c_l, \end{aligned} \tag{29}$$

where $\lambda = [\lambda_1, \lambda_2, \ldots, \lambda_{|\mathcal{L}|}]$. The Lagrange dual function is then given by

$$\begin{aligned} g(\lambda) &= \max_{\boldsymbol{\gamma_I}} L(\boldsymbol{\gamma_I}, \lambda) \\ &= \sum_{s\in\mathcal{S}} \max_{\gamma_{I,s}} \left[U_s(\gamma_{I,s}) - \tau \gamma_{I,s} \lambda^s \right] + \sum_{l\in\mathcal{L}} \lambda_l c_l, \end{aligned} \tag{30}$$

where $\lambda^s = \sum_{l\in\mathcal{L}(s)} \lambda_l$.

Each Lagrange multiplier λ_l, which can be interpreted as the delay at link l [50], is implicitly updated as

$$\lambda_l(t+1) = \left[\lambda_l(t) - \alpha \left(c_l - x_l^*(\lambda(t)) \right) \right]^+ \tag{31}$$

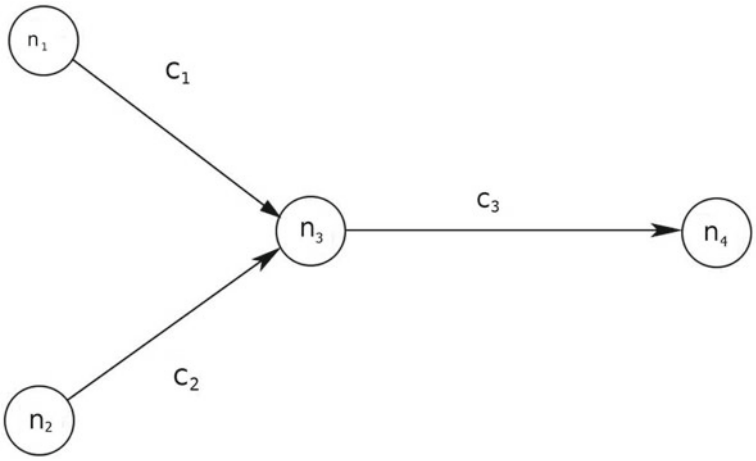


Fig. 17 Simple network topology demonstrating congestion between two video streams

where $x_l^*(\lambda(t)) = \sum_{s \in S(l)} \tau_s \gamma_{I,s}^*$ represents the total optimal rate offered at link l and α denotes the step size. We define $[\cdot]^+$ as $max(0, \cdot)$.

Since $U_s(\gamma_{I,s}) - \tau_s \gamma_{I,s}$ is differentiable, $\max_{\gamma_{I,s}} \left[U_s(\gamma_{I,s}) - \tau_s \gamma_{I,s} \lambda^s\right]$ is obtained when

$$\frac{dU_s(\gamma_{I,s})}{d\gamma_{I,s}} = \tau_s \lambda^s, \tag{32}$$

which states that the derivative with respect to the sampling rate should be equal to a scaled version of the delay. Since $U_s(\gamma_{I,s})$ (as defined in (25)) is a concave monotonically increasing function in $\gamma_{I,s}$, $\frac{dU_s(\gamma_{I,s})}{d\gamma_{I,s}}$ is decreasing in $\gamma_{I,s}$. Therefore, as λ^s varies, the optimal update direction of $\gamma_{I,s}$ is the *negative of the direction of the change in round trip time*.

The simplest interpretation of $\Delta RTT^i(t+1)$ as calculated in (24) and used in (27) for source i is the difference between consecutive delay measurements $\lambda^i(t) - \lambda^i(t-1)$. The update direction of $\gamma_{I,s}$ is then given by $(-\Delta RTT)$, which is the direction of the update in (27). Finally, it was shown in [51] that given a small enough step size, a gradient projection algorithm such as (27) will converge to the optimal sampling rate allocation.

Numerical simulations were also run to support this interpretation. Two simple networks were tested as shown in Figs. 17 and 18, respectively, where C_i represents the capacity on link i and N_i represents node i. The arrows represent video streams. In both cases, the optimal rate allocation was determined by solving the optimization problem directly as a semidefinite program using SeDuMi [46] with the convex optimization toolbox CVX [45], and the same problem was solved using the iterative algorithm (27).

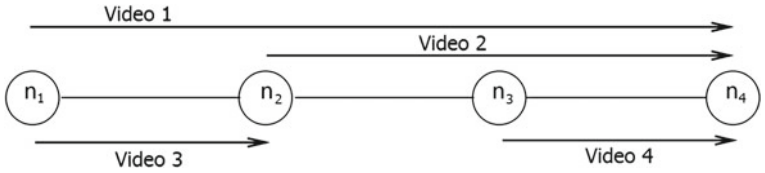


Fig. 18 Linear network topology demonstrating varying congestion between four video streams

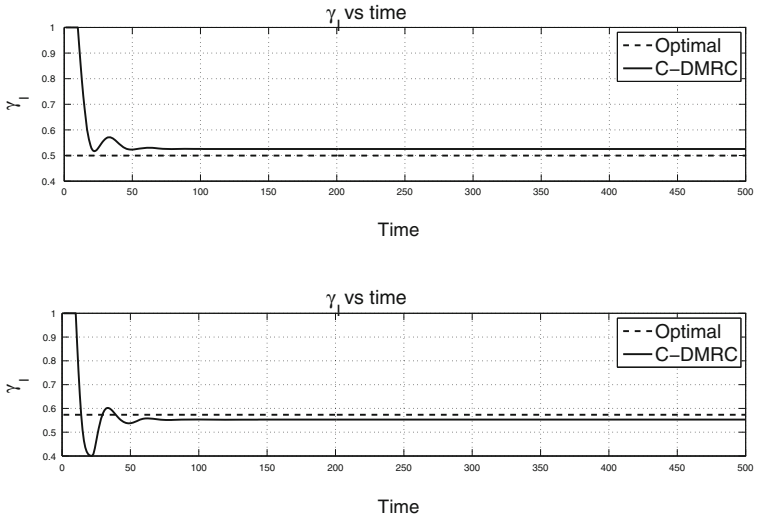


Fig. 19 Sampling rate from the C-DMRC rate controller compared to the optimal sampling rate allocation for topology 1

These two topologies were chosen because they verify two important requirements for a distortion-based rate controller. The network in Fig. 17 has two video streams with a single bottleneck link. This topology can be used to assure that two different videos with different rate-distortion properties achieve the same received video quality. The other topology, shown in Fig. 18, was used to show that the rate controller will take advantage of unused capacity. Video 3 in this network is only contending with a single other video, while the other three videos are contending with each other resulting in a higher optimal rate for video 3.

The results from these tests are shown in Figs. 19, 20, 21 and 22. Figures 19 and 20 show the *I* frame sampling rate of the videos compared to the optimal value, and Figs. 21 and 22 show the actual video qualities. In all cases, the rate found by the iterative algorithm was within 5 % of the optimal value as determined by the convex solver. The 5 % difference between the optimal rates and the rates obtained from the iterative algorithm are due to the step size of the algorithm. If the step size were decreased, the resulting rate would be closer to the optimal. However, making the step size too small results in an algorithm which is infeasible to implement because

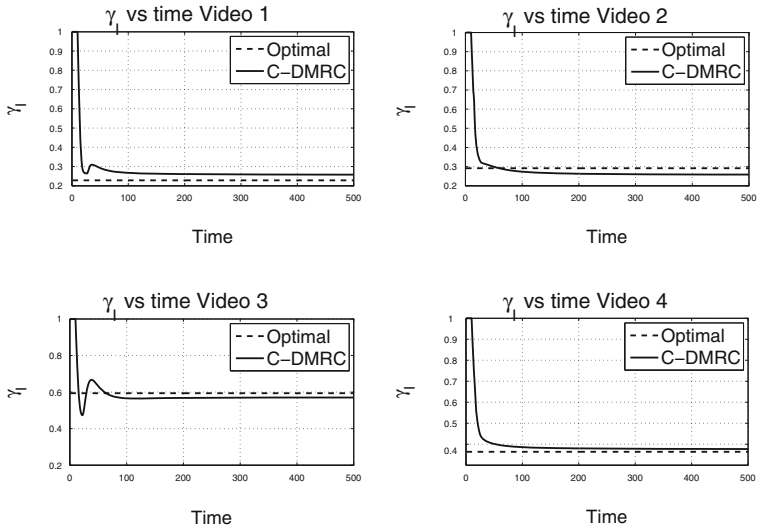


Fig. 20 Sampling rate from the C-DMRC rate controller compared to the optimal sampling rate allocation for topology 2

of the amount of updates needed. Finally, to avoid the trivial solution of all rates and qualities being equal, different videos were transmitted. The simulations show that the iterative algorithm achieved all requirements, and was nearly optimal for both networks.

9 Future Research Challenges

While the current research in the application of CS techniques is promising, there are still a few challenges that need to be solved before this technology can be realized in a realistic network. In this section, we will introduce some of these challenges.

9.1 Reconstruction Complexity

This paper has been focused on the sampling and encoding of video using compressed sensing. However, the biggest hurdle in CS reconstruction is the complexity required to reconstruct a video. Currently, the most common reconstruction algorithms are either least absolute shrinkage and selection operator (lasso) [52] or gradient projection for sparse reconstruction (GPSR) [38]. Others commonly seen are orthogonal matching pursuit (OMP) [53], stagewise orthogonal matching pursuit (StOMP) [47], basis pursuit denoising (BPDN) [54], and many others (see for example [55]). While

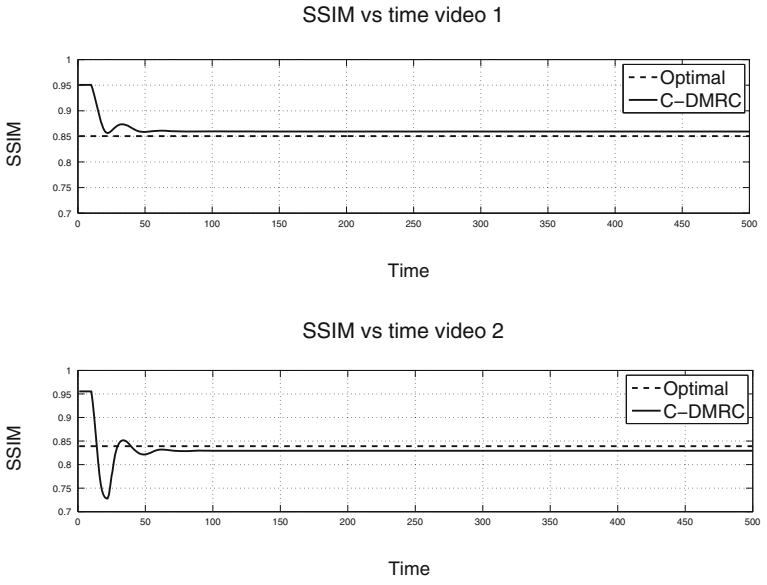


Fig. 21 Video quality from the C-DMRC rate controller compared to the optimal sampling rate allocation for topology 1

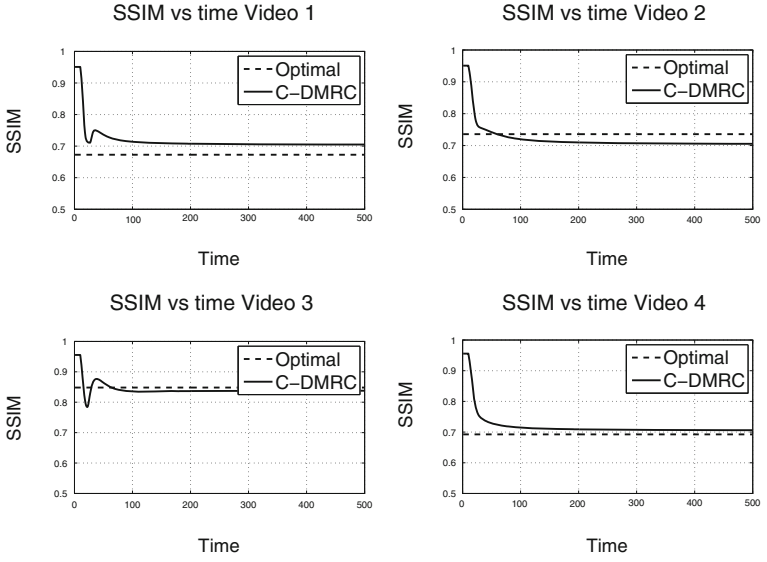


Fig. 22 Video quality from the C-DMRC rate controller compared to the optimal sampling rate allocation for topology 2

some of these algorithms are very fast, none of them can reconstruct video in real time, i.e., at 30 frames per second (or even 12 frames per second).

There are a few techniques for accomplishing this that may be promising. First is reducing the dimensionality of the signal, and reconstructing it in blocks, as is done in [56–58]. As stated above, since the complexity of even the fastest algorithm is (much) more than linear, reconstructing four $\frac{N}{2} \times \frac{N}{2}$ images will be faster than reconstructing one $N \times N$ image. However, the "amount" of sparsity in an image is related to the *size* of the image. As the image size decreases, the number of samples needed to reconstruct that image increases for the same reconstruction quality. This limits the practical applications of this technique.

Another processing technique for reducing the complexity is to use properties of the images in the reconstruction. For example, in [54], the authors present a scheme for iteratively updating a CS solution based on a previous solution. Since natural images are smooth, the difference in the sparse transforms of each column vector of an image can itself be represented as a sparse vector. This sparse column difference vector is then used to update the reconstruction of the previous column. The authors show that this system is indeed faster that others available. However, it is still not fast enough for real-time decoding.

9.2 Adaptive Sampling Matrices

A major issue in CS encoding of images is that, while the compression is good, it does not generally compare to more deterministic video compression methods. While we have shown that the power required to compress and transmit a video using CS techniques is much lower than traditional methods [59], reducing the compressed size of the video would present more applications for this technology.

One way to do this is to adapt the sampling matrix to the image, and increase the sparsity at the source. For instance, the sampling matrix $\boldsymbol{\Phi}$ and the sparse transform matrix $\boldsymbol{\Psi}$ in (2) can be specifically chosen to optimize the rate-distortion performance at each frame. While there are some rather obvious techniques to accomplish this, to be practical, the system must be able to adapt to the properties of the video *without first sampling the entire video*. The system must be able to work on a single pixel camera or similar device.

10 Conclusions

We have presented an introduction to compressed sensing as applied to video encoding. The goal of this chapter was to make a case for why CS should be used in video encoding for low power WMSN nodes. Currently available state-of-the-art algorithms are not suitable for sensor networks, and CS solves many of the problems associated with traditional methods. We have presented the background necessary

to begin approaching this problem. We have also described some of the leading algorithms developed for applying CS to video.

Acknowledgments This paper is based upon work supported in part by the National Science Foundation under grant CNS1117121 and by the Office of Naval Research under grant N00014-11-1-0848.

References

1. I. Akyildiz, T. Melodia, K. Chowdhury, Wireless multimedia sensor networks: applications and testbeds. Proc. IEEE **96**(10), 1588–1605 (2008)
2. S. Pudlewski, T. Melodia, A distortion-minimizing rate controller for wireless multimedia sensor networks. Comput. Commun. (Elsevier) **33**(12), 1380–1390 (2010)
3. I.F. Akyildiz, T. Melodia, K.R. Chowdhury, A survey on wireless multimedia sensor networks. Comput. Netw. (Elsevier) **51**(4), 921–960 (2007)
4. S. Soro, W. Heinzelman, A survey of visual sensor networks. Adv. Multimedia **2009**, Article ID 640386 (2009)
5. A. Kansal, S. Nath, J. Liu, F. Zhao, SENSE-WEB: an infrastructure for shared sensing. IEEE MultiMedia **14**(4), 8–13 (2007)
6. A.T. Campbell, N.D. Lane, E. Miluzzo, R. Peterson, H. Lu, X. Zheng, M. Musolesi, K. Fodor, S.B. Eisenman, G.S. Ahn, The rise of people-centric sensing. IEEE Internet Comput **12**(4), 12–21 (2008)
7. Advanced Video Coding for Generic Audiovisual Services. ITU-T Recommendation H.264 (2005)
8. T. Wiegand, G.J. Sullivan, G. Bjntegaard, A. Luthra, Overview of the H.264/AVC video coding standard. IEEE Trans. Circ. Syst. Video Technol. **13**(7), 560–576 (2003)
9. J. Ostermann, J. Bormans, P. List, D. Marpe, M. Narroschke, F. Pereira, T. Stockhammar, T. Wedi, Video coding with H.264/AVC: tools, performance, and complexity. IEEE Circ. Syst. Mag. **4**(1), 7–28 (2004)
10. T.Wiegand, G.J. Sullivan, J. Reichel, H. Schwarz, M. Wien, Joint Draft 11 of SVC Amendment. Doc. JVT-X201 (2007)
11. I.S. Reed, G. Solomon, Polynomial codes over certain finite fields. J. Soc. Ind. Appl. Math. **8**(2), 300–304 (1960)
12. J. Hagenauer, Rate-compatible punctured convolutional codes (RCPC codes) and their applications. IEEE Trans. Commun. **36**(4), 389–400 (1988)
13. E.J. Candes, Compressive sampling. in *Proceedings of the International Congress of Mathematicians* (Madrid, 2006)
14. D. Donoho, Compressed sensing. IEEE Trans. Inf. Theory **52**(4), 1289–1306 (2006)
15. E. Candes, J. Romberg, T. Tao, Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. IEEE Trans. Inf. Theory **52**(2), 489–509 (2006)
16. E.J. Candes, J. Romberg, T. Tao, Stable signal recovery from incomplete and inaccurate measurements. Commun. Pure Appl. Math. **59**(8), 1207–1223 (2006)
17. E. Candes, T. Tao, Near-optimal signal recovery from random projections and universal encoding strategies? IEEE Trans. Inf. Theory **52**(12), 5406–5425 (2006)
18. K. Gao, S.N. Batalama, D.A. Pados, B.W. Suter, Compressed sensing using generalized polygon samplers. in *Proceedings of Asilomar Conference on Signals, Systems and Computers* (Pacific Grove, 2010), pp. 1–5
19. K. Gao, S.N. Batalama, D.A. Pados, Compressive sampling with generalized polygons. IEEE Trans. Signal Process. **59**(10), 4759–4766 (2011)

20. M. Wakin, J. Laska, M. Duarte, D. Baron, S. Sarvotham, D. Takhar, K. Kelly, R. Baraniuk. An architecture for compressive imaging. in *Proceedings of IEEE International Conference on Image Processing* (*ICIP*) (2006), pp. 1273–1276
21. D. Takhar, J. Laska, M. Wakin, M. Duarte, D. Baron, S. Sarvotham, K. Kelly, R. Baraniuk, A new compressive imaging camera architecture using optical-domain compression. in *Proceedings of SPIE Conference on Computational Imaging IV* (San Jose, 2006), pp. 43–52
22. Crossbow Imote2 Mote Specifications (2010), http://www.xbow.com
23. Digital compression and coding of continuous-tone still images—requirements and guidelines. ITU-T Recommendation T.81 (1992)
24. N. Ahmed, T. Natarajan, K.R. Rao, Discrete cosine transform. IEEE Trans. Comput. **C-23**(1), 90–93 (1974)
25. D. Huffman, A method for the construction of minimum-redundancy codes. Proc. IRE **40**(9), 1098–1101 (1952)
26. S. Pudlewski, T. Melodia, On the performance of compressive video streaming for wireless multimedia sensor networks. in *Proceedings of IEEE International Conference on Communications*(*ICC*) (Cape Town, 2010)
27. K. Stuhlmuller, N. Farber, M. Link, B. Girod, Analysis of video transmission over lossy channels. IEEE J. Sel. Areas Commun. **18**(6), 1012–1032 (2000)
28. IEEE 802.15 WPAN Task Group 4 (TG4) (2012), http://grouper.ieee.org/groups/802/15/pub/TG4.html
29. Specification of the bluetooth system—version 1.1b, specification volume 1 & 2. Bluetooth SIG (2001)
30. Ieee std 802.11b-1999/cor 1–2001. (2001)
31. Ieee std 802.16-2004. (2004)
32. A. Graps, An introduction to wavelets. IEEE Comput. Sci. Eng. **2**(2), 50–61 (1995)
33. A. Bruckstein, D. Donoho, M. Elad, From sparse solutions of systems of equations to sparse modeling of signals and images. SIAM Rev. **51**(1), 34–81 (2007)
34. J. Romberg, Imaging via compressive sampling. IEEE Signal Process. Mag. **25**(2), 14–20 (2008)
35. S. Boyd, L. Vandenberghe, *Convex Optimization* (Cambridge University Press, New York, 2004)
36. I.E. Nesterov, A. Nemirovskii, *Interior-Point Polynomial Algorithms in Convex Programming* (SIAM, Philadelphia, 1994)
37. USC Signal and Image Processing Institute. The USC-SIPI image database (2012), http://sipi.usc.edu/database
38. M.A.T. Figueiredo, R.D. Nowak, S.J. Wright, Gradient projection for sparse reconstruction: application to compressed sensing and other inverse problems. IEEE J. Sel. Topics Signal Process. **1**(4), 586–598 (2007)
39. S. Pudlewski, T. Melodia, A. Prasanna, Compressed-sensing-enabled video streaming for wireless multimedia sensor networks. IEEE Trans. Mobile Comput. **99**, 1 (2011)
40. JPEG2000 Requirements and Profiles. ISO/IEC JTC1/SC29/WG1 N1271 (1999)
41. W. Sweldens, The lifting scheme: a new philosophy in biorthogonal wavelet constructions. in *Proceedings of the SPIE Wavelet Applications in Signal and Image Processing III*, Vol. 2569, ed. by A.F. Laine, M. Unser (1995), pp. 68–79
42. L. Gan, T. Do, T.D. Tran, Fast compressive imaging using scrambled block hadamard ensemble. in *Proceedings of European Signal Processing Conference* (*EUSICPO*) (Lausanne, 2008)
43. M. Duarte, M. Davenport, D. Takhar, J. Laska, T. Sun, K. Kelly, R. Baraniuk, Single-pixel imaging via compressive sampling. IEEE Signal Process. Mag. **25**(2), 83–91 (2008)
44. Valiux. DLP Digital Light Processing by Texas Instruments (2012), http://www.vialux.de/HTML/en_dlp.htm
45. M. Grant, S. Boyd, CVX: Matlab software for disciplined convex programming, version 1.21 (2012), http://cvxr.com/cvx
46. J. Sturm, Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. Optim. Methods Softw. **11**, 625–653 (1999)

47. D.L. Donoho, Y. Tsaig, I. Drori, J.-L. Starck, Sparse solution of underdetermined linear equations by stagewise orthogonal matching pursuit (Stanford Technical report, 2006)
48. Z. Wang, A. Bovik, H. Sheikh, E. Simoncelli, Image quality assessment: from error visibility to structural similarity. IEEE Trans. Image Process. **13**(4), 600–612 (2004)
49. S. Chikkerur, V. Sundaram, M. Reisslein, L.J. Karam, Objective video quality assessment methods: a classification, review, and performance comparison. IEEE Trans. Broadcast. **57**(2), 165–182 (2011)
50. S.H. Low, L.L. Peterson, L. Wang, Understanding TCP Vegas: a duality model. J. ACM **49**, 207–235 (2002)
51. S. Low, D. Lapsley, Optimization flow control I basic algorithm and convergence. IEEE ACM Trans. Netw. **7**(6), 861–874 (1999)
52. R. Tibshirani, Regression shrinkage and selection via the lasso. J. Roy. Stat. Soci. Ser. B (Methodol.) **58**(1), 267–288 (1996)
53. J. Tropp, Greed is good: algorithmic results for sparse approximation. IEEE Trans. Inform. Theory **50**(10), 2231–2242 (2004)
54. M. Asif, J. Romberg, Dynamic updating for ell_1 minimization. IEEE J. Sel. Topics Signal Process. **4**(2), 421–434 (2010)
55. Rice DSP Compressive Sensing Resources (2012), http://dsp.rice.edu/cs
56. T. Do, Y. Chen, D. Nguyen, N. Nguyen, L. Gan, T. Tran, Distributed compressed video sensing. in *Proceedings of IEEE International Conference on Image Processing* (*ICIP*) (2009), pp. 1393–1396
57. V. Stankovic, L. Stankovic, S. Cheng, Compressive video sampling. in *Proceedings of the European Signal Processing Conference* (*EUSIPCO*) (Lausanne, 2008), pp. 2–6
58. A. Wani, N. Rahnavard, Compressive sampling for energy efficient and loss resilient camera sensor networks. in *Proceedings of IEEE Conference on Military Communication* (*MILCOM*) (Baltimore, 2011)
59. S. Pudlewski and T. Melodia, "A Rate-Energy-Distortion Analysis for Compressed-Sensing-Enabled Wireless Video Streaming on Multimedia Sensors", in Proc. of IEEE Global Communications Conference (GLOBECOM), Houston, TX, December 2011

Chapter 15
Body Sensor Networks for Activity and Gesture Recognition

Narayanan C. Krishnan and Sethuraman Panchanathan

Abstract The last decade has witnessed a rapid surge of interest in new sensing and monitoring devices for health care applications. An important development in this area is that of Body Sensor Networks (BSN) that operate in a pervasive manner for on-body applications. Intelligent processing of the sensor streams from BSN is key to the success of applications that rely on this framework. In this chapter we dwell upon one application of BSN that involved processing of wearable accelerometer data for recognizing ambulatory or simple activities and activity gestures. We elaborate on the different steps such as feature extraction and classification involved in the processing of raw sensor data for detecting activities and gestures. We also discuss various aspects associated with a real-time simple activity recognition system such as computational complexity and factors that emerge considering that the sensors are worn by humans. While some of these factors are common to wireless sensor networks in general, the discussion of the chapter is focused on the BSN system developed by us for recognizing simple activities and activity gestures.

N. C. Krishnan conducted the work presented in this chapter while he was a PhD student at Arizona State University.

N. C. Krishnan (✉)
School of Electrical Engineering and Computer Science,
Washington State University, Pullman, WA 99163, USA
e-mail: ckn@eecs.wsu.edu

S. Panchanathan
Center for Cognitive Ubiquitous Computing, Arizona State University,
Tempe, AZ 85287, USA
e-mail: panch@asu.edu

H. M. Ammari (ed.), *The Art of Wireless Sensor Networks*,
Signals and Communication Technology, DOI: 10.1007/978-3-642-40066-7_15,

1 Introduction

Advances in the area of sensors, low power integrated circuits and wireless communication networks have enabled the development of a new generation of wireless sensor networks—Body Sensor Networks. Body Sensor Network (BSN) also referred to as Body Area Networks is a specific category of wireless sensor networks that are intended to operate in a pervasive manner for on-body applications. Much of the theory relating to general wireless sensors also relate to BSN and issues such as power optimization, battery life performance and radio design are key issues. Aspects that are unique to BSN include usability, durability, and robustness, how well the sensor fits in with the application, reliability, and security of data.

This sensor architecture has seen increasing interest over the last decade as it facilitates inexpensive and continuous monitoring of health of individuals. A number of intelligent physiological sensors can be easily integrated into a wireless body network that can be used for computer assisted rehabilitation of early detection of medical conditions. Understandably, most of the applications of BSN are drawn from the health care domain and include monitoring the physical activities of older adults for detecting wandering, falls and other behavioral changes, continuous monitoring, and tracking of patients suffering from chronic diseases such as diabetes, asthma and heart attacks. Other applications of BSN include sports—to understand the performance and kinematics of athletes; military—monitor the movements of soldiers and understand their activity; and for monitoring the lifestyle and general well-being of individuals for designing pro-active intelligent environments.

In this chapter, we discuss one application of BSN that concerns the recognition of ambulatory movements of individuals and activity gestures involved in complex activities such as cooking and taking medicine. Ambulatory movements such as walking, running, sitting, etc., that are predominantly defined by movements belong to the category of simple activities and the term—*Simple Activity Recognition* refers to the process of inferring activities defined by movements from sensor streams. BSN that consist of wearable accelerometers provide one of the best platforms for sensing movement information about these activities. Simple activity recognition system finds use in many applications in the health care domain. Systems that monitor the ambulation of an elderly individual to detect a fall or other abnormal movement patterns depend on the recognition of the simple activities. With the growing popularity of mobile devices embedded with inertial sensors, there has been a lot of focus on developing applications that can track the energy expenditure of an individual on simple physical activities. Recently there has also been a lot of focus on recognizing simple activities through inertial sensors for assisting the process of location estimation in GPS denied environments. In addition to health care applications, recognizing simple activities or other related movement patterns also find a lot of interest in gaming applications. An example of this is the popular Nintendo Wii mote gaming systems.

Recognition of these activities is not a trivial problem. Figure 1 illustrates biaxial accelerometer data for some simple activities. The movement in these activities can

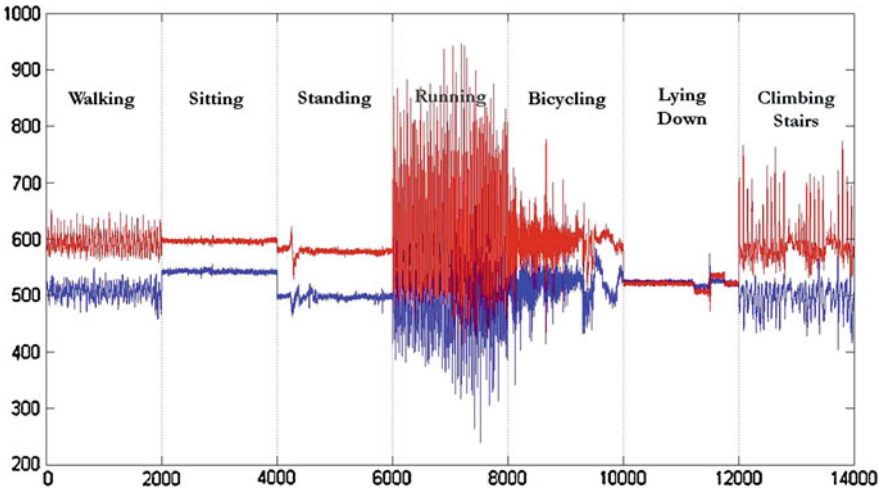


Fig. 1 Biaxial accelerometer data for different simple activities from an accelerometer placed on the ankle

be either characterized by a static posture as in the case of sedentary activities such as sitting and standing or is defined by a repetitive movement pattern as in the case of walking or bicycling. The different ways in which the continuous data stream from accelerometers can be modeled has resulted in different recognition paradigms. The goal of this chapter is to introduce the reader to field of wearable accelerometer-based activity recognition and to this end, the chapter describes the different aspects involved in designing a system that can detect simple activities in real-time and for recognizing activity gestures. It begins by discussing simple activity recognition approaches found in the literature in Sect. 2. It discusses the steps involved in the process of deciphering activities from raw sensor data. Section 3, then presents experiments conducted on the recognition paradigm on a publicly available simple activity dataset. These experiments lay the foundation for the real-time system that we have developed. The lessons learned from these experiments are then used to design and develop a simple activity recognition system that is presented in Sect. 4. Section 5 discusses wearable accelerometers for activity gesture recognition. Conclusion and pointers to some of the state of the art and future directions are highlighted in Sect. 6.

2 Background and Related Work

Much of the ongoing research focuses on prototyping wearable systems using BSN (accelerometers, microphones, pressure sensors, etc.) to recognize human activities. Miniaturized accelerometers have received attention from the bioengineering

community as an effective tool to monitor the physical activity of an individual [1, 2]. Accelerometers have also been used as an alternative modality for capturing motion information for recognizing ambulatory movements [3, 4]. With the increasing use of accelerometers for understanding human movements, different approaches have been adopted by researchers for analyzing and classifying the acceleration data streams, with some focusing on analyzing the data to extract salient features [5], while others concentrating on pattern recognition routines for detecting specific activities [6–9]. Since the focus of this chapter is on accelerometer-based activity recognition, we begin with a brief overview of an accelerometer.

Accelerometer is probably one of the most popular and ubiquitous sensors. It is has been made popular by devices and applications that change their functionality by changes in tilt or orientation such as the iPhone, or capture gaming devices that capture movements of individuals to interact with the game such as the Wii mote. An accelerometer is a sensor that measures the linear acceleration that is induced by gravity or by the movement of the sensor. It is sensitive to shock, orientation, and vibrations. An accelerometer is designed around the principle that a mass in acceleration exerts force. If the exerted force and the mass of the body can be measured, acceleration can be derived based on principles of physics (force = mass $\times$ acceleration). There are different kinds of accelerometers based on its type of construction and its sensitivity range. Most commonly found accelerometers are piezoelectric sensors that use the piezoelectric effect to measure the dynamic changes in the exerted forces. Some of the other categories of accelerometers based on its type of construction are MEMS, strain gauge, and capacitive. The sensitivity of an accelerometer is defined in terms of the acceleration due to gravity (g). Low g accelerometers can sense up to $\pm$12g accelerations, medium g accelerometers sense in the range of $\pm$100g, while high g accelerometers offer $>$100g sensing capabilities. Acceleration values in human movements typically fall in the range of Low g accelerometers.

Advances in MEMS technology has enabled the development of miniaturized accelerometers that can be worn by individuals. These sensors transform into powerful wearable sensing units when they are coupled with small form factor wireless communication technologies. There are a number of commercially available wireless accelerometers as presented in Fig. 2. Experiments in this chapter were conducted on data captured from two commercially available accelerometers—the WiTilt from Sparkfun [22] and ZStar from Freescale [23]. Other commercially available products are Mercury from Shimmer [24] and Wocket from MIT [25]. In contrast to an accelerometer, gyroscopes measure the angular velocity with respect to the inertial frame of reference. They are often used in conjunction with an accelerometer for activity recognition.

The process of deciphering an activity from the raw acceleration data involves a number of steps namely; *Preprocessing*, *Segmentation*, *Feature Extraction*, *Classification* and *Post Processing*. The subsequent paragraphs briefly describe these steps.

Preprocessing consists of steps that prepare the data for analysis, like the removal of high frequency noise spikes using techniques like nonlinear, low-pass median filters [3], Laplacian filters [10], and Gaussian filters [11] or the removal of the

Fig. 2 Some of the common wireless accelerometers available off-the-self. The images have been adapted from [22–26]

gravitational acceleration component from the data using high pass filters [1, 3, 12]. *Segmentation* plays an important role in systems that perform continuous activity recognition. It results in either fixed length or variable length segments depending on the set of target movements. Fixed length segmentation has been typically used for ambulatory motion recognition [6, 9, 13]. Explicit segmentation techniques resulting in variable length segments have been used [14] for spotting ambulatory movements and for recognizing workshop activities using audio features [15]. Holger et al. [16] uses a modified version of the SWAB algorithm to perform segmentation using the accelerometer and gyroscope data.

Feature extraction involves deriving salient and distinguishable features from the raw data. The statistical and spectral properties of the acceleration signal carry important cues for distinguishing different movements. Foerster et al. [1] and Knight et al. [17] employ only the mean value of the acceleration to distinguish ambulatory movements. The correlation between the axes of accelerometers placed at different locations in the body is another important feature for classification as illustrated by Bao et al. [13]. Although the effectiveness of other statistical features like variance, zero crossing rate, and mean crossing rate have been investigated by Maurer et al. [2] the performance of these features varies with respect to the activities.

The frequency content of the acceleration signal, represented as Fourier transform coefficients [5, 18] or aggregate quantities like spectral energy and entropy [13], have been used to differentiate activities like running and bicycling, or distinguishing between different walking patterns. Wavelets have also been used for detecting a variety of activities [10] and for modeling falls [19]. In addition to the features

extracted from the raw data, features from derivatives of the signal have also been explored for classification [6, 16, 20]. Huynh et al. [21] proposes to use multiple Eigen spaces for reducing the dimension of the acceleration data feature space for activity recognition.

Classification involves learning the mapping between the extracted feature vectors and the corresponding activity labels. Literature is abundant with techniques that use simple threshold-based models such as decision trees and decision tables and generative classifiers such as Hidden Markov Models (HMM) for recognizing simple activities. These techniques have been used in conjunction with a number of inertial sensors for simple activity recognition. Fixed length template matching using k-Nearest Neighbor (k-NN) has been quite popular among researchers for classifying simple activities using accelerometer data. Foerster et al. [1] employ k-NN for recognizing ambulatory movements using templates defined by the features extracted from fixed length window frames of accelerometer data placed at four different locations on the body. Jafari et al. [27] describes a method to detect the transitions between ambulation again using feature-based templates in conjunction with k-NN. Maurer et al. [2] also investigates the performance of k-NN for recognizing ambulation. All of these approaches first extract features from the acceleration movement pattern, which are in-turn used as the templates for matching.

For its computational efficiency naive Bayes classifiers are commonly used for recognizing ambulatory movements as demonstrated by Kern et al. [28], Bao et al. [13], Ravi et al. [9], and Maurer et al. [2]. The fixed length feature vectors extracted from movement data are used for learning the probability distribution of samples belonging to different classes. The probability distribution function is typically defined as a uni-modal Gaussian. Allen et al. [29] and Ibrahim et al. [5] use Gaussian mixture models for recognizing the transitions between different ambulation and walking patterns. These approaches assume that movement patterns of a specific type adhere to a Gaussian distribution. Olguin and Pentland [30] explore HMM for recognizing ambulatory movements from inertial sensor data obtained from different on-body locations.

Ambulatory movements consist of static body postures such as sitting, standing, and lying down along with dynamic movements involved in walking, running, climbing stairs, etc. Techniques that determine thresholds on features such as variance or energy, offer a simple approach for classifying these different types of movements. Thus decision trees such as C4.5 are a natural choice for accelerometer-based ambulatory movement recognition. Bao et al. [13] experiment with decision tables and trees for classifying around 20 different activities from accelerometer data collected from five different locations on the body. Karantonis et al. [12] discuss a real-time human movement classifier using tri-axial accelerometer using binary decision trees for classifying walking patterns. Maurer et al. [2] uses decision trees to recognize ambulatory movements using data from multiple sensors. Tseng and Cook [31] evaluate the performance of decision trees for determining the age of an individual based on movement patterns. Ravi et al. [9] experimented boosting and bagging with a number of base classifiers such as decision tables, decision trees, etc., for recognizing ambulatory movements from a single accelerometer and gyroscope readings placed

on the waist of an individual. Lester et al. [7] uses Adaptive Boosting (AdaBoost) to extract probabilities for classifying activities using data from multiple modalities. They also use AdaBoost as a feature analyzer to determine the importance of the different modalities for activity recognition. Artificial neural networks have also been used for activity and gesture recognition. Jafari et al. [27] explores MLP for learning the transitions between different ambulation. Yang et al. [32] discuss a time-delay neural network-based approach for recognizing motion patterns from trajectories extracted from image sequences. Mantyla et al. [33] recognizes static hand gestures from accelerometer data using self-organizing maps (SOM). Laerhoven et al. [20] also use SOM in conjunction with a Markov chain for differentiating ambulatory movements using data from a single inertial sensor placed on the thigh of the individual.

Suutala et al. [8] use support vector machines (SVM) for classifying ambulatory movements using accelerometers located on the thigh, wrists, and neck regions. Ravi et al. [9] use SVM along with boosted SVM for recognizing different activities using a single accelerometer placed on the waist of an individual. They observe that boosted SVM offers only a marginal improvement in classification performance over a regular SVM. Tseng and Cook [31] illustrate the effectiveness of SVM over other classification techniques such as multi-layer perceptron (MLP) and decision trees for determining the age of an individual using movement patterns. While these are the different approaches adopted in the literature for simple activity recognition, the performance of discriminative classifiers for simple activity recognition has not been explored to a fuller extent. The work presented in this chapter is motivated by the need to evaluate powerful discriminative classifiers for simple activity recognition. In particular it presents experiments conducted with Adaptive Boosting, Support Vector Machines, Regularized Logistic Regression, and Hidden Markov Models. A description of each of these techniques is provided in the following paragraphs.

AdaBoost, short for Adaptive Boosting, is a meta-machine learning algorithm, formulated by Freund and Schapire [73]. It is typically used in conjunction with many other learning algorithms such as C4.5 decision trees and naïve Bayes to improve their performance. AdaBoost is adaptive in the sense that subsequent classifiers built are tweaked in favor of those instances misclassified by previous classifiers. Thus as the iterations progress, AdaBoost increases the weights of those instances that have been consistently misclassified by the previous classifiers thereby increasing the likelihood of learning a classifier in the subsequent iteration that can correctly classify this instance. As a result, AdaBoost is sensitive to noisy data and outliers. It has however shown good improvement in the performance over the base classifiers for a number of applications. The classifiers it uses can be weak (i.e., display a substantial error rate), but as long as their performance is not random (resulting in an error rate of 0.5 for binary classification), they will improve the final model. The final model is a linear combination of the weak classifiers. Even classifiers with an error rate higher than by chance will be useful, since they will have negative coefficients in the final linear combination of classifiers and hence behave like their inverses. While one can use different base classifiers in conjunction with AdaBoost, we use binary decision stumps with the objective of building a simple model. We request

the reader to refer to the seminal article of Freund and Schapire [73] for more details about AdaBoost.

A support vector machine [74] is a very popular supervised learning method that analyzes data and recognizes patterns, used for classification and regression analysis. The standard SVM takes a set of input data and predicts, for each given input, which of two possible classes forms the input, making the SVM a non-probabilistic binary linear classifier. Given a set of training examples, each marked as belonging to one of two categories; an SVM training algorithm constructs a hyperplane or a set of hyperplanes in the high-dimensional feature space. It selects the hyperplane that achieves good separation between the data points of the two classes. This optimal hyperplane has the largest distance to the nearest training data point of both the classes. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on. One of the common tricks used with SVM is that of mapping the data points belonging to space where they are not linearly separable to a higher dimensional space where the separation becomes easier. This mapping is usually performed using the Kernel function, which reduces the computational complexity of the algorithm, by calculating the distance between the points in the high-dimensional space instead of determining the actual mapping of the points. There have been many commonly used Kernel functions such as polynomials, radial basis functions and sigmoid functions. The choice of the kernel function and the parameters of the function are dependent on the data and the application domain.

While AdaBoost and SVM are non-probabilistic classifiers, regularized logistic regression is a generalized linear model that follows binomial regression for deriving classification probabilities for each of the data sample. Regularized Logistic regression (RLogReg) [74] is a probabilistic discriminative approach for the classification problem. The posterior probability of a class is written as a logistic sigmoid function of the feature vector. The maximum likelihood estimation process is employed to determine the coefficients of the logistic sigmoid function. Regularization is performed to derive small values for the coefficients of the logistic sigmoid function. The margin of classification obtained from AdaBoost and SVM can be used to approximate the posterior probability of a class through the logistic regression process.

Hidden Markov models (HMM) are probably the most commonly used generative models for modeling movement patterns. It is a powerful generative model that includes a hidden-state network. HMM is rich in mathematical structures; it serves as the theoretical basis for a wide range of applications. It can model spatio-temporal information in a natural way. It also has elegant and efficient algorithms for learning and recognition, such as the Baum-Welch algorithm and Viterbi search algorithm. HMM has attracted a lot of attention as a useful tool for modeling the spatio-temporal variability in gestures. The unique internal segmentation property of the HMM i.e., the states and transitions of a trained HMM represent sub-patterns of a gesture and their sequential order, makes it a popular choice for modeling varying length spatio-temporal patterns. HMMs are effective for modeling temporal data because the global features extracted from the data do not explicitly capture the temporal nature of the data. Another advantage of HMMs is these samples can be of varying length.

Post Processing consists of smoothing the classification results [6, 7] or correcting the label based on domain constraints as illustrated By Suutala et al. [8].

3 Experimental Framework

This section discusses the feature extraction, classification and post classification label-smoothing techniques that have been employed leading up to the real-time system for simple activity recognition.

3.1 Description of the Dataset

The data used for conducting experiments with the computational framework for simple activities is a subset of the data collected by Bao et al. [13]. The data was collected in two different ways—supervised approach (activity), where the subject is given explicit instructions about what action to perform, and a semi-naturalistic approach (obstacle), where the subject is given instructions to perform an activity of daily life that implicitly encodes the action patterns. The data corresponding to 10 random subjects from a pool of 20, for 7 lower body activities namely *walking, sitting, standing, running, bicycling, lying down* and *climbing stairs*, from accelerometers placed at *hip, dominant ankle, and non-dominant thigh*, for the two modes of data collection have been considered for the experiments performed in this work. The data was collected from biaxial accelerometers that were strapped to the different body locations using Velcro. The accelerometers are sampled at approximately 76.25 Hz. Figure 1 depicts typical samples that are obtained from the accelerometers.

3.2 Feature Extraction

The first step in the feature extraction process is to divide the acceleration stream into frames. The acceleration stream is divided into frames of size 512 samples, with 256 overlapping samples between successive frames, as described by Ravi et al. [9]. For each frame, the statistical features like mean, variance, correlation between all the axis of all the accelerometers, along with the spectral features like energy and entropy are computed. Figure 3 illustrates the projection of these features onto a three dimensional space derived through principal component analysis. For activities that have a significant amount of motion like walking, running, etc., the rate of change of acceleration is a characteristic property that distinguishes them. These variations are captured by computing statistical features like mean, variance and correlation between all the axes on the first order derivative of the acceleration data in addition to the features mentioned above.

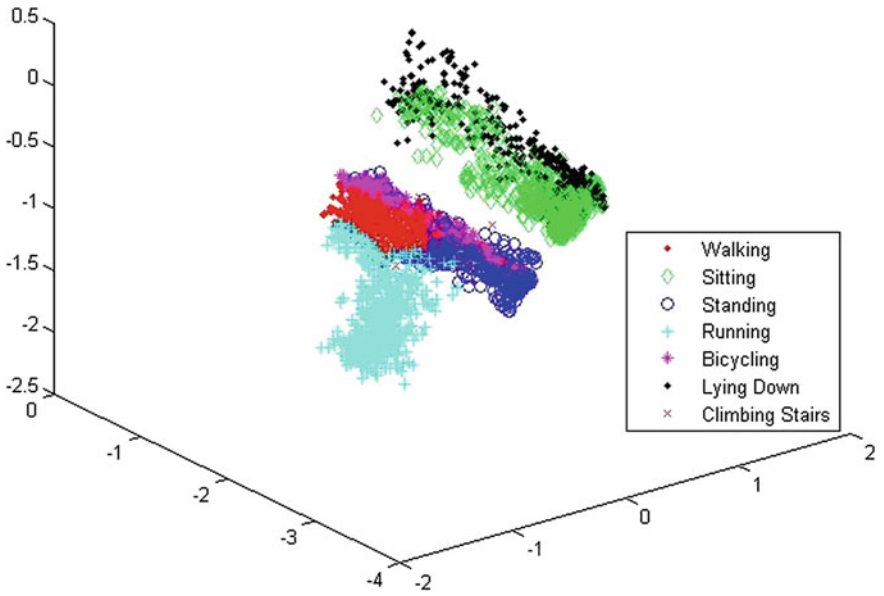


Fig. 3 Projection of the simple activity data samples onto three dimensions obtained through principal component analysis

The effect of different features on classification performance of AdaBoost was studied for determining the relevance of the features for discriminating simple activities. Separate AdaBoost classifiers were trained with the standard set of features, statistical features of the first derivative of the acceleration data and a combination of both. The accuracies for the three scenarios were 89.82, 81.9 and 92.81 %, respectively. It is evident from Fig. 4 that the standard features perform significantly better than the derivative features. However, there was a 34% increase in the accuracy when both features were combined. Figure 4 presents the class-wise accuracy for the three scenarios. It can be noticed that the derivative features are able to distinguish accurately activities characterized by distinctive motion patterns like walking, running, etc., (1, 4, 5, and 7). The accuracies for these classes are on par with that of the standard features. This indicates that features extracted from the first derivative of acceleration data are able to capture the subtleties in the motion data.

3.3 Isolated Recognition

The performance of AdaBoost, SVM, and RLogReg on the features extracted from each frame was evaluated for developing the computational framework for isolated simple activity recognition. Binary classifiers were trained for each activity. The AdaBoost classification routine was implemented in MATLAB based on the

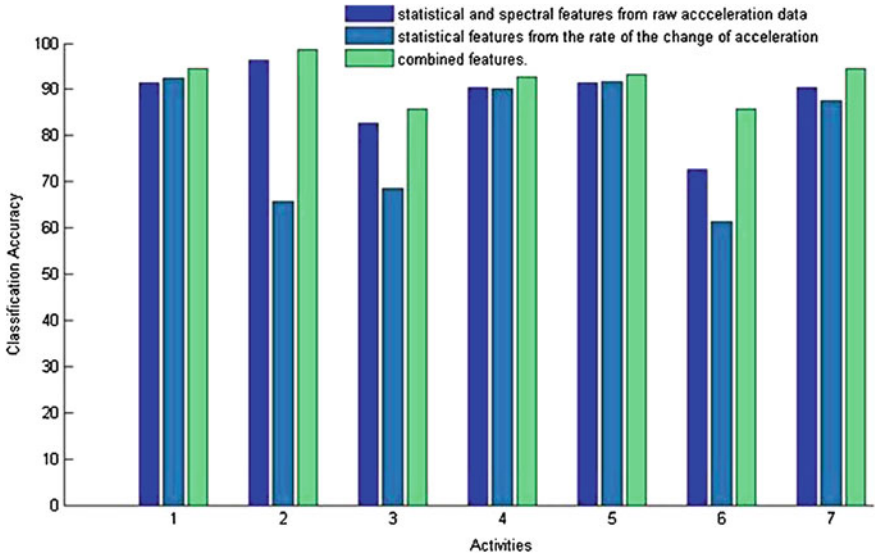


Fig. 4 Class-wise accuracies using AdaBoost trained on the three features (*1* walking, *2* sitting, *3* standing, *4* running, *5* bicycling, *6* lying down, *7* climbing stairs)

Table 1 Subject independent, adaptive, and dependent accuracies for the different discriminative classifiers

Classifiers	Subject independent	Subject adaptive	Subject dependent
AdaBoost	92.81	93.96	47.88
RLogReg	86.55	88.14	74.56
Linear SVM	82.28	83.60	72.64

description of the algorithm in Duda and Hart [34]. The SVM and RLogReg implementation from the SVMLight package by Joachims [35] and Komarek's Logistic regression toolbox [36] was used respectively. Given a test sample, the class that yielded maximum margin/probability, was considered as the predicted activity.

Three different evaluation scenarios were considered for the analysis. For the *subject independent* scenario, activity data from nine subjects were considered as training samples and the obstacle data from the remaining one subject was the test data. The activity data of all the ten subjects were considered as the training set and the obstacle data from each of the subject formed the test set, for the *subject adaptive* scenario. The activity and obstacle data from only a single subject formed the training and test set for the *subject dependent* evaluation.

The results summarized in Table 1 show that AdaBoost performed best in both subject independent and adaptive scenario, while RLogReg had the highest accuracy in subject dependent case. The 90 % reduction in the size of the training data for the subject dependent scenario was the cause for the poor performance

Table 2 The aggregate confusion matrix obtained from subject independent 10 folds cross validation using AdaBoost trained on combined features

Activity	Walking	Sitting	Standing	Running	Bicycling	Lying down	Climbing stairs
Walking	840	0	18	2	9	0	20
Sitting	0	296	0	0	0	4	0
Standing	0	13	128	0	8	0	1
Running	10	0	6	458	11	0	9
Bicycling	0	0	32	0	443	0	0
Lying down	0	57	0	0	0	343	0
Climbing stairs	13	0	4	1	1	0	323

of AdaBoost. We did not experiment with kernels for SVM due to the high computational costs associated with them. The confusion matrix for classification aggregated over the 10 subjects for subject independent scenario using AdaBoost is presented in Table 2. The misclassification of walking samples as climbing stairs and vice versa, suggests that the motion patterns involved in them are similar. There were also misclassifications occurring between activities that do not involve any quantitative motion in them, probably indicating that necessity of data from other parts of the body. A probable reason for this is that, data from accelerometers placed only in the lower parts of the body have been considered. Adding data from other accelerometers might improve the classification of these activities. Misclassification of bicycling samples as standing seems to be a very strange anomaly. On further analysis of the data, it was observed that these samples did not show any representative motion pattern associated with bicycling, thus can be possibly considered as outliers.

3.4 Post Classification Label Smoothing for Continuous Recognition

Human activity is a continuous process and though these discriminative techniques are effective in classifying an individual frame, they do not consider temporal continuity for classification. Based on the results from the previous experiments we observed that a number of samples that have been misclassified were actually in the midst of a continuous stream of correctly classified samples. This means that the strict condition of independent samples can be relaxed in this scenario to correct some of the inaccuracies in classification. Lester et al. [7] and Suutala et al. [8] propose to use a Hidden Markov Model trained on the probability of classification obtained through AdaBoost and SVM, respectively, to correct these types of errors. In this section a classification framework that incorporates this temporal continuity of human activity is proposed that does not require re-computation of the feature vector nor require any additional training, thus remains computationally inexpensive. It relies on the sim-

ilarity of successive samples in the continuous stream to combine the probabilities of classification.

Formally, classification margin $m_c(f_t)$, for a frame f_t, belonging to a class c derived either in AdaBoost or SVM reflects the confidence of prediction. This margin can be used by the classifier to output the probability, $p_c(f_t)$, of the frame belonging to class c. A method to compute the probability directly is to fit a sigmoid function to the output of AdaBoost or SVM as described in the following equation.

$$p_c(f_t) = \frac{e^{\phi m_c(f_t)}}{1 + e^{\phi m_c(f_t)}}$$

ϕ is a constant that is determined empirically through a cross validation process on the training dataset. The probability values computed for the frame f_i at time instant i can aid in classifying successive temporally close frames. For a frame f_t, let the frames that influence its classification be f_i, where $i = t - \Delta t, \ldots, t$. We weighed the probability $P_c(f_i)$, for the frame at i belonging to class c, by two factors - a function of i (temporal distance between the frames) denoted by $g(i)$ and a function of the similarity between the current frame and the past frame, measured as the Euclidean distance between them denoted by $h(t - i, t)$. Thus the final probability $P_c(f_t)$ for the frame at t, is given by the following equation where the denominator acts as a normalizing factor.

$$P_c(f_t) = \frac{p_c(f_t) + \sum_{i=1}^{\Delta t} g(i) * h(t - i, t) * P_c(f_{t-i})}{\sum_{i=1}^{\Delta t} g(i) * h(t - i, t) + 1}, \quad c = 1 \ldots 7$$

For the experiments conducted in this work, the function $g(i)$ was treated as a Gaussian distribution. This was done to ensure that frames that are farther away in time have minimal influence on each other. The function $h(t - i, t)$ was represented as $h(t - i, t) = e^{\{-\alpha d(f_{t-i}, f_t)\}}$, where $d(\cdot)$ corresponds to the Euclidean distance between the feature vectors describing the frames. This assumes that if adjacent frames are similar, then they should belong to the same class. This framework is illustrated in Fig. 5. Though in this work, experiments were conducted with only AdaBoost and RLogReg, the proposed framework can be adopted to work with any classifier.

The continuous acceleration stream from the obstacle dataset as a sequence of overlapping frames was considered for evaluating the proposed methodology. The number of past frames considered for classifying the current frames was varied. The optimal performance was achieved when three past frames were considered for classifying the current frame. AdaBoost and RLogReg classification routines were considered for the evaluation of the framework. While adding temporal information to static AdaBoost resulted in an average 10 fold cross validation accuracy of 95.35 %, RLogReg resulted in 89.63 %. For both algorithms, an improvement of about 2.5–3 % was observed.

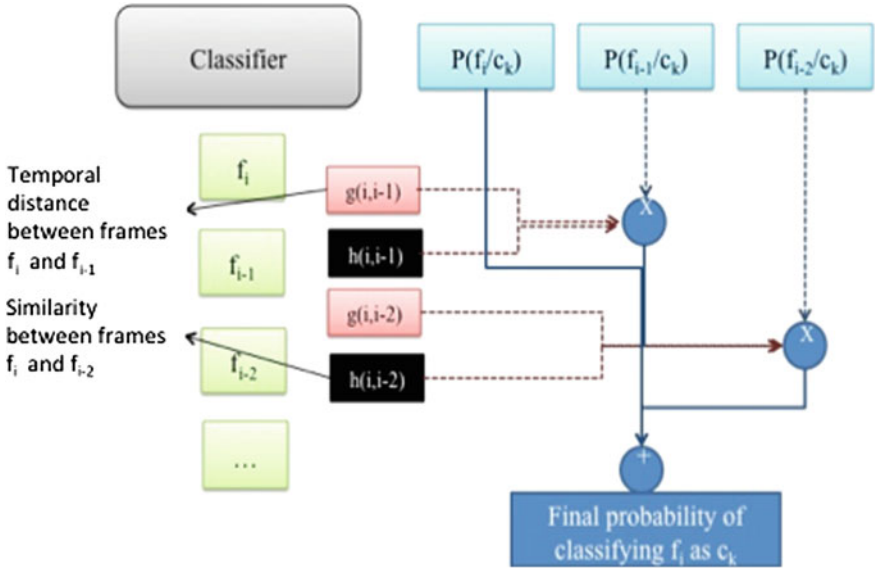


Fig. 5 Illustration of the post classification label-smoothing framework

3.5 Human Factors Assisting in Continuous Classification

The human body can be viewed as a kinematic system with well-defined degrees of freedom of movement. This kinematic system defines the transitions that can occur between different activities. For example, an individual cannot shift directly from a sitting state to a running state. The human body goes from a sitting state to a standing state and then to a running state. This sequence of states is actually reflected in the data captured through wearable sensors. A mechanism that can capture this information can be used to assist in the continuous classification process. The action grammar work of Ivanov and Bobick [37] is one way to capture this information. Ivanov and Bobick [37] in their seminal work on action grammars combine the syntactic and statistical schools of pattern recognition for the purpose of activity recognition. Statistical knowledge about the components or low-level primitives is combined with the structural knowledge expressed in the form of grammar. The syntactic knowledge acts as a constraint to the recognition of individual components as well as directs the process of recognition of the activity as a whole. The methodology proposed by them is useful in the context of recognizing the high-level activity.

In the context of simple activity recognition, this type of a grammatical framework can be used to validate the transitions between different activities. For example, a walking state cannot be reached from a sitting state without passing through the standing state. Thus the structure of the transition relationship between these activities can be modeled using action grammars. Adding these types of constraints has the potential to improve the performance in a continuous scenario. The framework

Table 3 Transition matrix for the CFG defined for simple activities

Activity	Walk	Stand	Sit	Run	Lie down
Walk	0.6	0.2	0	0.2	0
Stand	0.3	0.3	0.2	0.2	0
Sit	0	0.2	0.6	0	0.2
Run	0.2	0.2	0	0.6	0
Lie down	0	0	0.2	0	0.8

described in the previous section does not explicitly model the transitions between the activities and thus a context free grammar (CFG)-based approach may aid the recognition process.

While Ivanov and Bobick's work describe how to integrate an action grammar into a statistical recognition framework, they do not discuss how to build the grammar. It is assumed that the grammar exists or can be derived from the domain. In scenarios where the action is defined by a small group of gestures (such as the examples described in their work drawing a square), these models can be constructed manually. To validate the transitions between the different simple activities, a CFG was manually constructed based on the human constraints in the performance of these activities. The five simple activities that were considered for this study are sitting, standing, walking, running and lying down. A CFG that represented the relationship between the different activities was constructed. Probability values were assigned to each of the possible transition in the CFG. This probabilistic CFG can be represented in terms of a first order Markov chain transition matrix. The entries in the matrix were crafted manually by counting he transitions between the different activities in the training dataset. It was assumed that the tendency of a subject to remain in an activity was higher than moving to another activity. This is reflected in high probability values for the self-transition elements in the transition matrix. The probability values for all the other transitions are equally distributed. Table 3 describes the probability values of this transition matrix.

It is interesting to note from the transition matrix that the 'Stand' is the unstable activity amongst the rest as it can easily transit to other activities. The self-transition probability of this activity is lowest. This is in accordance with what one typically observes in the real world. This matrix defines finite state automata that can be used to smooth out the labels post classification. The data from the previous study was used to study the properties of such a finite state automata. Samples corresponding to cycling and climbing stairs were removed from the continuous stream of sensor data. The AdaBoost classifier was used for recognizing the individual samples. The classification probabilities for every time step, derived through boosting were the input to the finite state automata. The probability values were propagated through the automata and normalized for every time step. The state resulting in maximum normalized probability was considered as the activity label for that particular time step. The average error rates obtained from testing on data collected from 4 subjects were 0.08 and 0.09 with and without the finite state automata model. It can be seen that adding the CFG framework improves the accuracy only by a

marginal amount. The static AdaBoost model itself resulted in very high recognition rates, thus adding the grammatical framework does not influence the outcome significantly.

4 Real-Time Classification

Encouraged by the results obtained in the previous section, a real time system for detecting and recognizing lower body simple activities (walking, sitting, standing, running, and lying down) using streaming data from tri-axial accelerometers was designed. The first step is collecting data from the two accelerometers and passing this information through the preprocessing and segmentation stage. In this second stage, spurious noise in the data is removed and the continuous stream is broken down into more manageable segments. Each data segment then passes through the feature extraction step, where salient features are extracted to characterize the properties of the raw data. These fixed-length feature vectors are then sent through the classification stage, where a trained AdaBoost classifier is used to identify the activity corresponding to the sample and the probability of the classification is computed. The next sections describe each of the steps of the activity recognition system framework in more detail.

The unique contribution of the research presented here is the use of data gathered from a limited number of accelerometers. While the performance is marginally superior to the best of the previous results, this approach is distinct in using only accelerometer data. It improves on standard feature extraction frameworks by using a boosted classifier for recognition, resulting in a system that has a very high accuracy for real-time activity recognition.

4.1 Data Capture

The system relies on off-the-shelf accelerometers (WiTilt v2.5 employing a Freescale MMA7260Q triple axis accelerometer with class 1 bluetooth) connected to a computer using a wireless Bluetooth serial port. These accelerometers are sturdy and have only minimal data loss over long periods of continuous sampling ($<1\,\%$ for 600 s). Data from three subjects (two males and one female) was collected. The accelerometers were placed on the right ankle and on left thigh, with the x-axis facing perpendicular to the ground, to maintain consistency across the subjects as illustrated in Fig. 6. Each subject was asked to perform five different activities (walking, sitting, standing, running, and lying down) for duration of 1 min. The accelerometers were sampled at the rate of 100 Hz. The 60s trial of a subject was then broken down to smaller chunks each consisting of 100 acceleration samples. This gives a total of 180 chunks per activity combining the data from all the subjects. Each sample corresponds to the acceleration data from each of the three axes of the two accelerometers.

Fig. 6 Accelerometer placement for the real-time simple activity recognition system

4.2 Data Processing and Feature Extraction

The second step is to break the continuous data stream into equal-length segments of information, which is an approach proposed by Bao et al. [13]. In this framework, each segment consists of 100 samples and successive segments have an overlap of 50 samples. The sampling rate of the accelerometers is 100 Hz, so each segment corresponds to 1 second from the data stream. This time interval proved to be sufficient for analyzing the activities we were trying to recognize (walking, sitting, standing, running, and lying down). In the next step statistical and spectral features described in our prior work was extracted. These features are computationally inexpensive and characterize most of the distinguishing features for separating the activities considered for the real-time implementation (walking, sitting, standing, running, and lying down).

4.3 Classification

A threefold cross validation technique was used to learn the optimal activity model. For each fold, the data corresponding to one subject was used for the testing the model learned by training the AdaBoost on the remaining data. The classifier stabilized after 250 iterations. We used AdaBoost as it had resulted in the best performance in the set of experiments discussed before. The average subject independent accuracy obtained in this fashion was around 95.2 %. We also experimented with the data

Fig. 7 Illustration of the real-time system developed by us for recognizing simple activities. The data streams from the two accelerometers can be seen in the lower size of the application, with the activity classification probabilities plotted above. The current example shows the subject to be in the Walking phase

from each accelerometer: when data from only the ankle accelerometer was used, the accuracy of the classifier dropped to 62 %; when data from only the thigh accelerometer was used, the accuracy was 83 %. Analysis of the confusion matrix obtained after the classification revealed that data from the ankle accelerometer was insufficient for classifying two activities (sitting and standing), while data from the thigh accelerometer was insufficient for classifying four activities (sitting, lying down, walking, and running). Clearly, these experiments demonstrate the necessity of multiple accelerometers for recognizing the activities selected for our system. However, when one is interested in designing a system for a subset of these activities (for example, sitting, standing, walking and running), a single accelerometer (located in the thigh) might result in high classification accuracy. The choice of the location of the accelerometer is very much dependent on the activities that we want to recognize. There is no clear deterministic way of identifying the optimal location and one has to rely on empirical investigations for achieving this.

Finally, the data from all three subjects were used to train the classifier, with the resulting model incorporated into the continuous activity recognition framework. Figure 7 illustrates the system developed by us for recognizing the simple activities. This classifier has been presented in a number of live demonstrations using volunteers not in the training database illustrating its robustness and dependability. Figure 8 illustrates the real-time classification for an interesting example. The data consists of a total of approximately 8000 samples (corresponding to 80 s), and it can be seen that the classification is only 84.4 % accurate once the activity stabilizes. This particular subject walked faster than the three training subjects in the database and illustrates

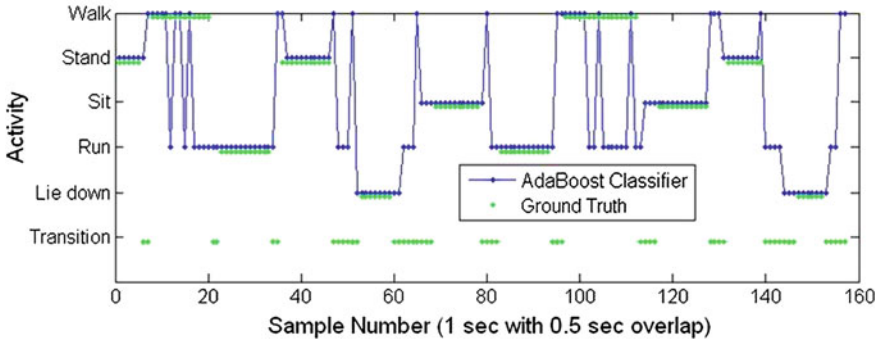


Fig. 8 An example of real-time continuous classification comparing the output of the AdaBoost classifier with the ground truth

the most likely type of misclassification (confusion between walking and running) that can occur using the current classifier. If walking is excluded as a possible ground truth, the accuracy of the system returns to our typical 98 %. It was also noticed that the probability of classifying the samples as walking was low. However, the threshold that was used to remove transitions and other arbitrary activities was not sufficient to detect these anomalies. This clearly indicates the need for an adaptive threshold model for detecting out-of-vocabulary samples.

4.4 Computational Complexity

In this section, the theoretical analysis of the computational complexity of the simple activity recognition system is first presented. The activity recognition system consists of two parts— feature extraction and classification. Thus the overall computational complexity CC of the system can be broken down as follows:

$$CC = CC_{FE} + CC_{CL},$$

where CC_{FE} corresponds to the complexity of feature extraction and CC_{CL} is the complexity associated with the classification step. The current real-time system consists of two accelerometers each sampled at 100 Hz. Continuous data from these two sensors were classified using sliding window protocol. The window was of length 100 samples (1 s) with an overlap of 50 samples between successive windows. Statistical and spectral features consisting of mean, variance, correlation energy, and entropy were extracted from each window. Multiple AdaBoost classifiers trained on samples for each of the activity was used to classify the sample.

Let us now look at the order of complexity of the feature extraction step. Table 4 lists the set of features considered in this work, along with their computational

Table 4 Computational complexity of the features extracted, where N is the number of data samples

Feature	Order	Dimensions
Mean	$O(N)$	6
Variance	$O(N)$	6
Correlation	$O(N^2)$	15
Spectral energy	$O(N \log N)$	6
Spectral entropy	$O(N \log N)$	6

complexity. Thus every sliding window requires a total of $12O(N) + 15O\left(N^2\right) + 12O(N \log N)$ computations for extracting the features. N refers to the number of accelerometer samples in each window. In the proposed system this value is 100. The total number of computations for extracting the feature vector is approximately 1.6 e06.

Focusing on the classification step, the AdaBoost classifier was trained using decision stumps. A decision stump is a primarily a threshold-based classifier and has a computational complexity of $O(1)$. The maximum number of boosting iterations during training was set to 100. It was observed that some of the classifiers had learnt less than 100 weak hypotheses. However, for computing the worst case scenario, assume that each of the classifier has 100 hypotheses. Thus the computational complexity of classifying a frame is 500 $(100 \times number\ of\ classes\ =\ 5)$.

Therefore, the total computational cost of extracting and classifying a single window can be derived as

$$CC(N) = 12O(N) + 15O(N^2) + 12O(N \log N) + 500O(1)$$

and in this particular example, for $N = 100$, this value turns out to be $\approx$ 1.6e06 computations. Assuming that the device needs D units of battery discharge to complete one computation, the discharge lost by the battery for classifying one window is 1.6 e06/D. If the complete discharge of the battery happens after T discharges, the duration of the battery in this system will be $\frac{T \times D}{1.6e\ 06}$s.

Boyd et al. [38] conducts an extensive evaluation on the effect of different parameters of an accelerometer-based activity recognition system by identifying Pareto-optimal points in the operational parameter design space. For a practical solution to the above question, an experiment similar to the one proposed by Boyd et al. [38] was designed to study the effect of variation in the key parameters of the system such as sampling rate, window size and features extracted on the performance of the system. The computational complexity of the system was measured in terms of normalized computation time for classifying one window. The performance of the system was measured in terms of its accuracy. Simple activity data from three subjects were used in this experiment. Figure 9 depicts the impact of these parameter variations on the accuracy of the system. The accuracy was the average value obtained through three rounds of subject independent evaluation. The sampling rates for the experiment were chosen to be 100, 50, 20, and 10 Hz. The window size values were 1, 2, 4, 6, 8,

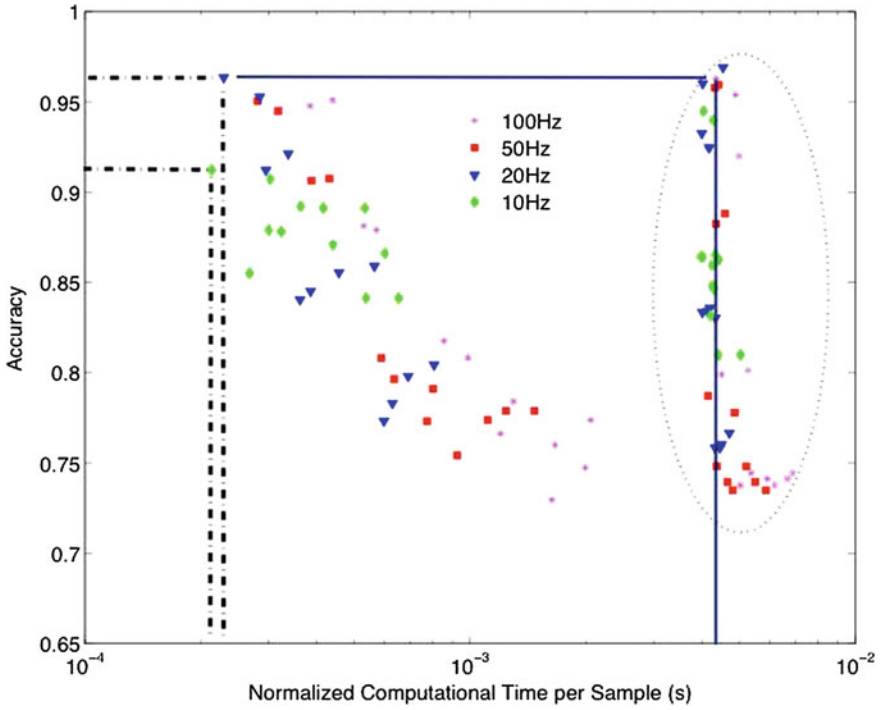


Fig. 9 Comparing the normalized computational time for different parameters of the system against the performance measured as accuracy

and 10 s. The features set considered in this system was divided into statistical (mean, variance, and correlation) + spectral (energy and entropy), statistical only, mean + spectral, mean + variance. Each of these combinations of operational parameters is represented as a point in the plot according to its normalized computational time and accuracy.

The points inside the ellipse correspond to the scenarios where correlation between the axes was used as a feature for classification. The distinct separation between these points and other points with respect to the computational time clearly supports the fact that correlation is an expensive feature. At the same time, it is clear that parameters without the correlation feature did not result in change in the performance of the system. Thus the contribution of correlation toward the performance of the system is significantly less compared to other features.

It is very interesting to note that the system is able to support low power consumption without compromising on the performance as indicated by the green diamond and the blue triangle in the top left corner of Fig. 9. The current parameters (indicated by the solid lines) are suboptimal in terms of the computational time. However, this can be rectified by using the operational parameters described by the points on the top left part of Fig. 9 without compromising on the accuracy. Assuming that one unit

of battery discharge takes place during the time required for the computation of a sample; the optimal parameters for the system would make the battery last 5 % longer than the current operational parameters.

5 Activity Gesture Recognition

Traditional, accelerometer-based activity recognition systems have been used to monitor and assess elderly individuals by detecting and recognizing the high-level ambulatory movements (walking, sitting, standing, climbing stairs, etc.) that are part of the activities of daily life. The steady posture or the repetitive movement that defines the simple activities facilitates easy and reliable recognition as discussed in the work presented in the previous chapter. However, complex activities such as making a drink or cooking consists of various complex, short duration movements (predominantly hand movements)—activity gestures, along with many interactions with objects. Tracking these complex activities relies on reliable recognition of both the objects and the activity gestures.

The primary aim of this section is to support multi-modal systems for tracking the complex activity rather than to develop an activity recognition system based solely on activity gestures. There are many approaches in the literature where the overall activity of an individual is determined using sensors embedded in the environment such as RFID tags or reed switches. The problem with these approaches is that, while they provide high-level information about the activity such as making a drink or making brownie, by gathering information about the objects, they do not reliably detect the tasks involved in these activities. For example, given that an individual is interacting with a spoon, kettle, tea bags, and water, it is easy to reliably infer that the activity is making tea by exploiting object-activity relationships. But mere object information is insufficient for recognizing the tasks involved in the activity such as pouring milk, scooping sugar, stirring tea. That is, holding a spoon in hand provides information about the possible tasks that the person is doing with the spoon such as scooping sugar, stirring milk, or mixing tea, but cannot pin point which among these is he/she actually performing. This fine level of information is required to effectively predict the future tasks that an individual might perform. The movement information is essential for understanding the activity task. While it is probably not feasible to differentiate these gestures, from movements in other activities of daily living; given the context that the current activity is making a drink, is it possible to detect and recognize the activity gestures is the problem addressed here. The context can be derived through other sensing mechanisms using information such as the location of the individual, or objects of interaction and is assumed to be given in this work. This section of the chapter describes an approach for recognizing activity gestures from wearable accelerometer data.

5.1 Gesture Recognition approaches

The recognition of isolated gestures has been studied extensively over years and many approaches have been proposed to tackle the diverse problems. Given a gesture/activity sample, these approaches classify it as belonging to one of the labels. They do not attempt to detect where the activity or gesture began or ended in a continuous stream. In general, these approaches can be broadly divided into three categories based on the computational framework adopted for classification: Template based, generative and discriminative approaches. In the next three sub-sections a review of the different approaches in these three categories are discussed.

5.1.1 Template-Based Approaches

Template matching approaches are transductive in nature and can be divided into two categories based on the type of the templates: fixed length and variable length. Fixed length template matching using k-NN has been quite popular among researchers for classifying simple activities using accelerometer data. Foerster et al. [1] employ k-NN for recognizing ambulatory movements using templates defined by the features extracted from fixed length window frames of accelerometer data placed at four different locations on the body. Jafari et al. [27] describe a method to detect the transitions between ambulation again using feature-based templates in conjunction with k-NN. Maurer et al. [44] also investigate the performance of k-NN for recognizing ambulation. All of these approaches first extract features from the acceleration movement pattern, which in-turn are used as the templates for matching.

Variable length template matching is performed using a simple length-based normalization or approaches based on the more advanced dynamic time warping (DTW) as demonstrated by Sakoe et al. [45]. Corradini et al. [46] use DTW as a matching technique for determining the similarity between the unknown input and a set of previously defined templates for video-based sign language recognition. Darrell and Pentland [47] employ DTW for matching space-time motion trajectories associated with manipulating objects for recognizing isolated gestures. Niyogi and Adelson [48] and Gavrila and Davis [49] use DTW to match sequences of joint model configurations obtained from image sequences. Veeraraghavan et al. [50] propose a DTW method for action recognition that allows better modeling of variations within model sequences. More recently, Alon et al. [51] propose a constrained dynamic programming-based DTW approach for spotting gestures from continuous data streams. The approach proposed by them unifies the spatial and temporal characteristics of the sample for the purpose of matching. While the DTW-based approach is quite popular among the computer vision community, there are only a few approaches that employ DTW for accelerometer-based movement pattern recognition.

5.1.2 Generative Approaches

Generative models are used in machine learning for either modeling data directly (i.e., modeling observed draws from a probability density function), or as an intermediate step to forming a conditional probability density function. A conditional distribution can be formed from a generative model through the use of Bayes' rule. Generative models contrast with discriminative models, in that a generative model is a full probability model of all variables, whereas a discriminative model provides a model only of the target variable conditional on the observed variables. Examples of generative models include Gaussian mixture model, hidden Markov model and naive Bayes.

Starner and Pentland [52] implemented an HMM-based system for recognizing sentence-level American Sign Language (ASL) without explicitly modeling fingers from video sequences. Bregler et al. [53] learn a kind of switching-state HMM over a set of autoregressive models, each approximating linear motions of blobs in a video frame. Oka et al. [54] propose HMM-based recognition of gestures from motion trajectories of finger tips tracked in a video stream. Gandy et al. [55] use HMM for modeling hand gestures from motion trajectories determined using an IR camera. Lester et al. [7] employ HMM as a post processing tool to smoothen out the recognition results of an AdaBoost classifier for detecting human activities using data from on-body sensors. Olguin and Pentland [30] explore HMM for recognizing ambulatory movements from inertial sensor data obtained from different on-body locations. Ward et al. [15] use HMM on top of features obtained from Linear Discriminant analysis for recognizing activities in a workshop such as sawing, drilling, etc. Junker et al. [16] use HMM for recognizing gestures from body-worn inertial sensors in their two-stage user activity detection algorithm. Al-ani et al. [19] model walking speeds and transitions between different ambulation using a HMM. Mannil et al. [75] propose a methodology for rejecting irrelevant movements from the dataset using a Hidden Markov model. The goal of their approach is to filter irrelevant movements from relevant postural changes from data collected through wearable accelerometers.

Various extensions to the more general class of dynamic Bayesian networks (DBN) have been proposed to overcome limitations of HMM as illustrated by Ghahramani et al. [57]. Brand et al. [56] learn coupled HMMs to model interactions between several state variables. They use a two state coupled HMM to recognize interactions between left and right hand motions during Tai Chi exercises. Park and Aggarwal [58] use a complex DBN to model interactions between two persons, such as hugging, handshaking, and punching. Peursum et al. [59] model interactions between people and objects in their work using Bayesian networks. Nguyen et al. [60] propose to use hierarchical HMMs for video activity recognition. Jojic et al. [61] and Toyama and Blake [62] extend HMMs with separate latent states for posture and view for pose and view invariant activity recognition.

5.1.3 Discriminative Approaches

In contrast to their generative counterparts, discriminative approaches model the conditional probability of the target variable given the observations. These approaches typically model the boundary separating the data samples between two classes. While this genre of techniques is commonly used for a variety of classification and regression tasks, it is applicable to the problem of recognizing movement patterns, when the pattern is described in terms of a feature that implicitly encodes the spatio-temporal variation in the movement pattern.

Cui and Weng [63] discuss an appearance based multidimensional discriminant analysis approach for selecting linearly discriminating features for hand gesture recognition using a recursive partition tree approximator. Lester et al. [46] use AdaBoost to extract probabilities for classifying activities using data from multiple modalities. They also use AdaBoost as a feature analyzer to determine the importance of the different modalities for activity recognition. Ong et al. [64] propose a modified version of the AdaBoost algorithm using decision trees as weak learners for recognizing hand shapes (gestures) from video streams. Pentney et al. [65] discuss a virtual evidence-based boosting algorithm for recognizing activities by fusing accelerometer data with RFID tags. Yang et al. [66] use SVM for classification of static or short duration hand gestures for improving the overall performance of the American Sign Language recognition system. Morency and Darrell [67] discuss an intelligent user interface through head gestures classified using SVM.

A significant amount of recent work has shown the power of discriminative models such as conditional random fields (CRF) for specific sequence labeling tasks. CRF use an exponential distribution to model the entire sequence given the observation sequence [68–70]. This avoids the independence assumption between observations, and allows non-local dependencies between state and observations. A Markov assumption may still be enforced in the state sequence, allowing inference to be performed efficiently using dynamic programming. CRFs assign a label for each observation (e.g., each time point in a sequence), and they neither capture hidden states nor directly provide a way to estimate the conditional probability of a class label for an entire sequence. Sminchisescu et al. [70] applied CRFs to classify human motion activities (i.e., walking, jumping, etc.) and showed improvements over an HMM approach. When the sequence under consideration has distinct sub-structure, models such as hidden-state CRFs (HCRF) [71] that exploit hidden state is advantageous. Wang et al. [71] propose a gesture recognition technique based on HCRF, which can estimate a class given a segmented sequence. Since they are trained on sets of pre-segmented sequences, these HCRF models capture only the internal structure and not the dynamics between gesture labels. Morency et al. [72] proposed latent-dynamic hidden CRFs (LDCRFs) for vision based gesture recognition. LDCRFs are a framework for detecting and recognizing sequential data, which can model the sub-structure of a label and learn dynamics between labels.

One can thus see that there has been substantial amount of work in the literature on gesture recognition, which is predominantly vision based. There has not been much work in exploring the feasibility of these approaches for accelerometer-based gesture

Table 5 The mock and semi-naturalistic scenarios used for data capture

Activity gesture	Mock scenario (CS1)	Semi-naturalistic scenario(CS2)
Pour	Take the glass that is full and pour its contents into the empty glass. Pour a small quantity every time.	Pour the water from the glass.
Scoop	Use a spoon to scoop contents from the glass that is full into the empty glass	Use two scoops of powder for making the drink
Unscrew cap	Unscrew the lid of the water bottle. Pause for a couple of seconds. Screw on the lid on the bottle	Open the powder drink jar, and close it after you finish using it
Stir	Take the spoon and stir the contents of the glass for 30 s	Ensure the powdered drink has dissolved by stirring the mixture
Lift to Mouth	Take an empty glass and pretend that you are drinking water from the glass by taking several short sips	Drink the glass of beverage that was prepared

recognition. In this section we evaluate the performance of the classifiers from each of the three genres for recognition of activity gestures. The focus is on the recognition of gestures that build the basis for the inference of more abstract activities. The primary aim is to support complex activity spotting systems rather than to develop an activity spotting system based solely on hand gestures. Nonetheless, this work shows how, for suitable domains, good performance can be achieved without any additional information.

5.2 Data Collection

Data for five activity gestures namely—*lift to mouth, pour, stir, scoop*, and *unscrew cap* were collected using the wearable accelerometers. Two different scenarios were used for collecting these data. While a semi-naturalistic mode of collecting the data that permits a greater degree of freedom to the subject in performing the activity is proposed by Bao et al. [13], for the activity gestures considered in this work it is not practical for a subject to perform the same activity a number of times in one session. Instead, an alternative data capture session was devised during which the subjects enacted the same movements with mock objects a number of times, thereby providing sufficient data samples for training. For each of the five gestures, alternate scenarios representing the actual movement needed to perform the activity were designed. The mock scenarios are explained in Table 5. Explicit instructions were given on how to perform each movement. Each subject was asked to perform each of the actions 20 times. Figure 10 describes the settings for the data capture session, and a video

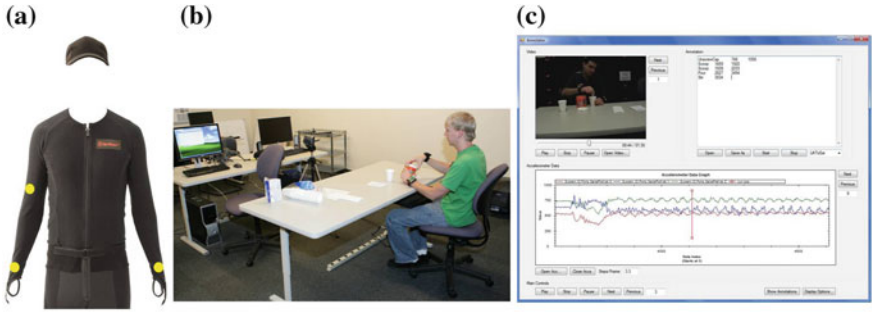


Fig. 10 **a** Illustrates the different locations for placing the accelerometer, **b** Depicts the data capture setup and **c** shows the annotation software that synchronizes the video and accelerometer data

camera was used to record the sessions. The subject started in a 'rest' state, where the hands rested on the table, and the subject began the action after receiving a cue from the experimenter. The video was used off line to synchronize the accelerometer data and to extract relevant portions from the continuous data stream that corresponded to the actions. In addition data was also collected from the subjects performing multiple trials of the actual activity in an unrestricted manner. This consisted of two sessions performed during different days. During each session the subjects were asked to make a glass of powdered drink and drink it, twice. The video recordings from these sessions were later used for annotating the actions.

All the subjects in our experiments were college students aged 22–28 years, and all of the subjects were right-handed. Three accelerometers were used for data collection. The first accelerometer was placed on the wrist of the right hand and the second accelerometer was placed just above the right elbow. We observed during our experimental data capture sessions that subjects sometimes used their left hand as support during the actions, so the third accelerometer was placed on the left wrist. Each of the accelerometer-sensing units was attached using Velcro tapes.

5.3 Feature Analysis

Figure 11 illustrates the samples collected from a single subject using the mock setup (activity gesture set CS1). These signals were obtained from the accelerometer placed on the right wrist. The most evident observations are that samples are of varying length and that each action can be distinguished by observing the acceleration patterns. For example, unscrewing the cap can be defined by a number of rapid repetitive movements, while slower repetitive movements represent stir. A dip in the z-axis acceleration appears for the actions scoop and lift to mouth, but the y-axis values increase for scoop and fall significantly for lift to mouth. Similar observations can be made for other actions, leading to the conclusion that it is possible to differentiate these actions using the accelerometer data we gathered.

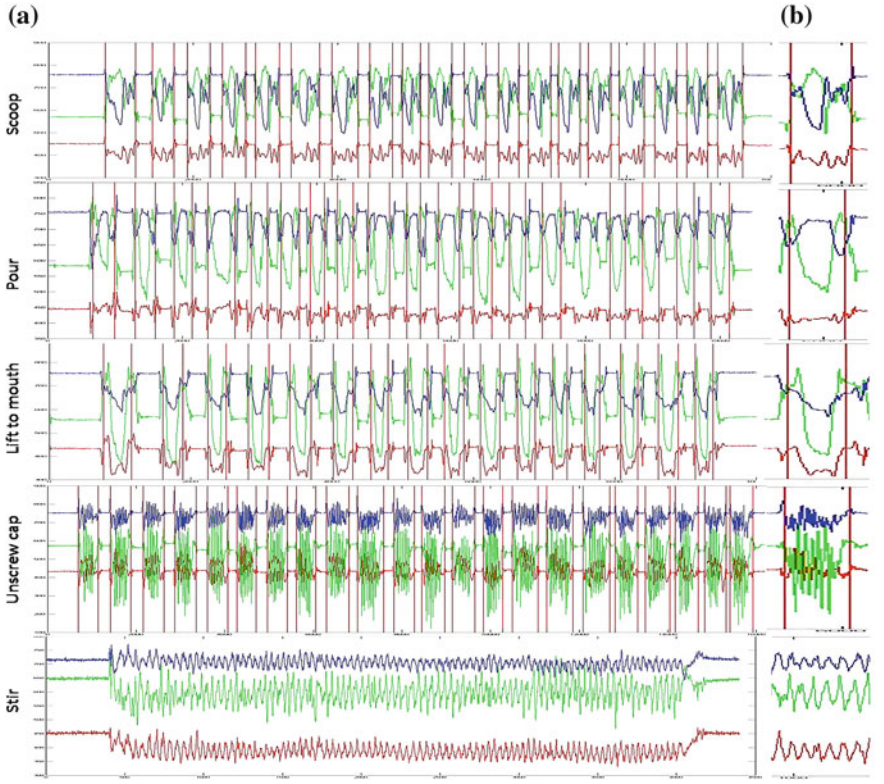


Fig. 11 The action samples collected from one subject using the accelerometer placed on the right wrist. The RGB lines stand for the x-, y-, and z- axis, respectively. The continuous stream of data was annotated offline using the video recording. **a** Samples recorded from a single data capture session. **b** One representative sample for each of the actions

To analyze these gestures, several statistical and aggregate spectral features: mean, variance, correlation, spectral energy, and spectral entropy were extracted. The spectral energy value was computed as the sum of the squared amplitude of discrete FFT coefficients, ignoring the DC coefficient. Similarly, the spectral entropy was computed as the normalized information entropy of the magnitudes of the discrete FFT coefficients, also ignoring the DC coefficient. Since the DC coefficient represents the mean of the signal, and is already being captured explicitly, we ignored it while computing these additional features. Each feature was computed for each axis of each accelerometer. Pair wise correlation between all the accelerometer axes (across all accelerometers) was also computed.

The feature analysis presented here is based on the data collected from the right wrist only. The problem of variable length sequences was circumvented by extracting fixed length feature vectors (mean: 6, variance: 6, correlation: 15, energy: 6, entropy: 6, totality of dimensions: 39) from the sequence. Similarity of samples from different

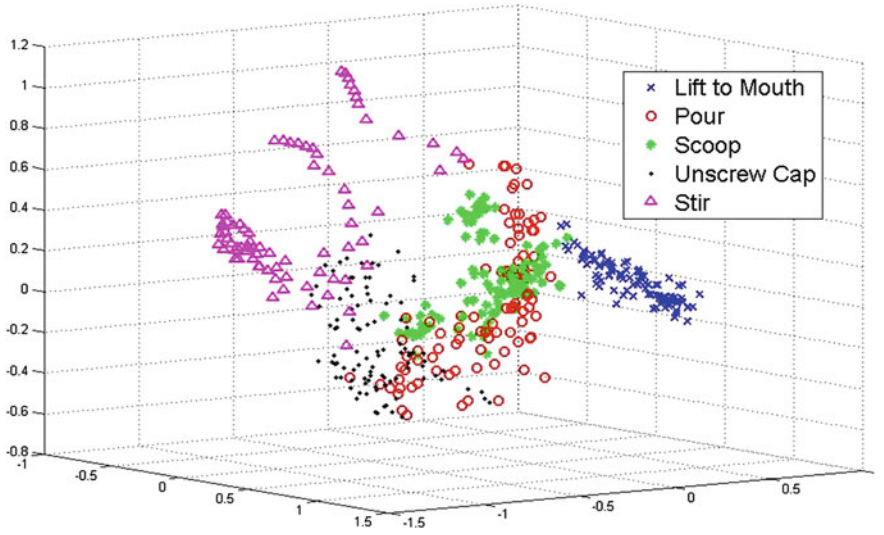


Fig. 12 Principal component analysis of activity gesture samples from all the subjects

subjects was visualized by conducting a principal component analysis (PCA) on the feature vectors. The distribution of the samples belonging to the different classes reduced to a 3-dimensional space is illustrated in Fig. 12. The action lift to mouth represented by the cross marks is distinct, but there was some overlap between the other actions (probably indicating that each subject performed the actions differently). Data points, belonging to the same class and the same subject, obtained after PCA, resulted in tight clusters. The multiple clusters belonging to the same class corresponded to different subjects.

These features were further analyzed using the AdaBoost framework. AdaBoost works by iteratively calling a certain weak learning algorithm known as the base learner to arrive at a classifier that gives better than 'by chance' accuracy, and constantly updates the distribution of weights of the samples after every iteration. If the base learner is a decision stump, then the process of boosting results in a number of weighted decision stumps, which when combined linearly gives the final class label. These weights associated with the decision stumps can further be used to derive weights for each of the feature dimensions. Feature dimensions with relatively high weights can then be considered to characterize the corresponding gestures more effectively. This process was adopted to study the importance of the different features extracted with respect to the activity gestures.

Based on the distribution of weights computed through AdaBoost, for every gesture, features with weights, twice or more than that of the uniform weights were selected. Table 6 presents the features that were selected through this process. It is interesting to note that with the exception of the gesture unscrew cap, all other gestures have nearly 10 dominant features (out of a total of 39) contributing to nearly 60 % of the total weight. The significance of these features was further

Table 6 Dominant features for each of the gestures as calculated using AdaBoost (X1, Y1, Z1 and X2, Y2, Z2 are the X, Y, and Z axes of the accelerometers placed on the wrist and elbow, respectively)

Activity gesture	Dominant feature	Aggregate weight
Lift to mouth	Mean (Z1), variance (X1, X2), correlation (X1 and Y1, Y1 and Z1, Y1 and X2)	57.59
Pour	Mean (X1), correlation (X1 and Z1, Y1 and Z1, Z1 and X2, Z1 and Z2, X2 and Z2)	65.88
Scoop	Mean (Y1, Z1, Y2), correlation (Y2 and Z2), entropy (Z1)	68.56
Unscrew cap	Mean (Z2), correlation (X1 and Y1, X1 and X2, Y1 and X2)	28.99
Stir	Mean (X2, Z2), variance (Y1), correlation (X1 and Z1), energy (Y1, Z1), entropy(Y1)	59.94

measured, by observing the classification performance by training AdaBoost only on these features. The AdaBoost trained on this reduced feature set resulted in an accuracy of 89.12 %, which was approximately the same accuracy obtained using all the features. This clearly validates the use of AdaBoost as a feature selection mechanism.

It was also noted that the features selected were intuitive in characterizing the gestures. As an example, for the activity gesture lift to mouth that is represented by the variations in the accelerometer data on the wrist, the majority of the features selected corresponded to this characterization. A similar observation was made with respect to all the other gestures as well, except for unscrew cap. While one would expect the energy and entropy features for the unscrew cap to have high significance due to the repetitive movement of the hand, low weight values were observed for these features. This probably also indicates a need to explore other features, for understanding some of these gestures.

5.4 Classification: Results and Discussion

The performance of k-NN a template matching model, HMM a generative model and AdaBoost a discriminative model was evaluated using three scenarios:

- Subject Independent Evaluation: This was performed using a leave one out strategy where data from four subjects was used as the training data to be tested on the fifth subject, in a round robin fashion. This evaluation provides the most difficult classification scenario for the classifiers.
- Subject Adaptive Evaluation: This was also performed using a leave one out strategy, with a modification. In addition to the data from four subjects, 25 % the data from the fifth subject was also used during training and the resulting classifier was tested on the remaining 75 % of the samples of the 5th subject.

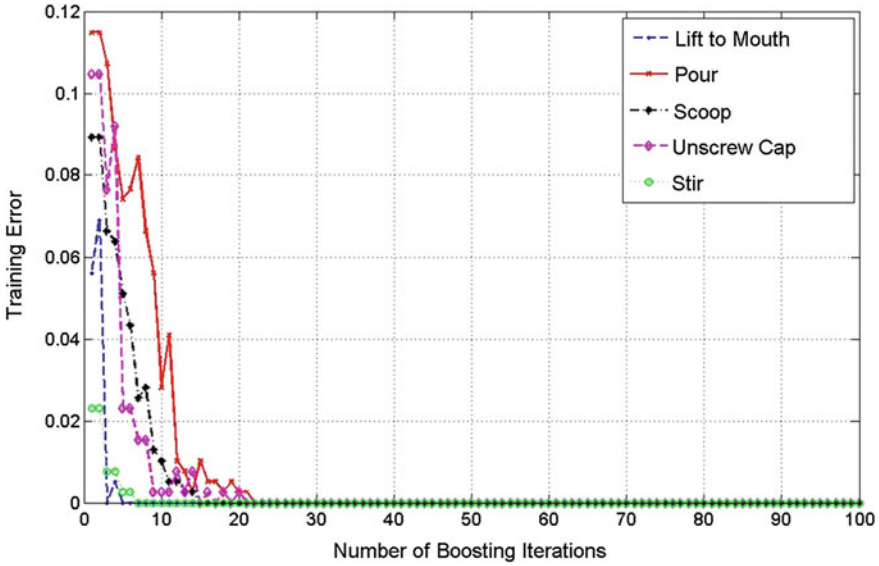


Fig. 13 Training error using AdaBoost across iterations for the different activity gestures

- Subject Dependent Evaluation: In this scenario, part of the data (50 %) from a subject was used for training and the remaining for testing. This can be correlated to a scenario where the system has been personalized for a single subject and is being evaluated for this subject alone.

AdaBoost had demonstrated excellent results in previous experiments with similar accelerometer data for ambulatory motion [52]. Each AdaBoost classifier passed through a maximum of 100 iterations. The training error reduced exponentially over the iterations and is illustrated in Fig. 13. It can be seen that there was a uniform decrease in the training error across all the actions.

As in the case of AdaBoost, a HMM was trained for each of the classes and the label with maximum likelihood estimated by the classifiers was chosen as the winner. The statistical and spectral features were extracted at a rate of 10 Hz (determined through empirical evaluation) from the data. This stream of features was used to train the HMMs. Each state in the HMM was modeled using a Gaussian mixture model. The parameters for HMM namely the number of Gaussians at every state and the number of hidden states—was decided using a trial and error method of cross validation. The optimal value for the number of hidden states and the number of GMMs was found to be 3 and 3, respectively. The same HMM parameters were used for training all the classes.

Since k-NN was a common technique used for classifying ambulatory movements, we also experimented with it for classifying hand movements. Dynamic Time Warping-based distance techniques that can handle data of varying length need exten-

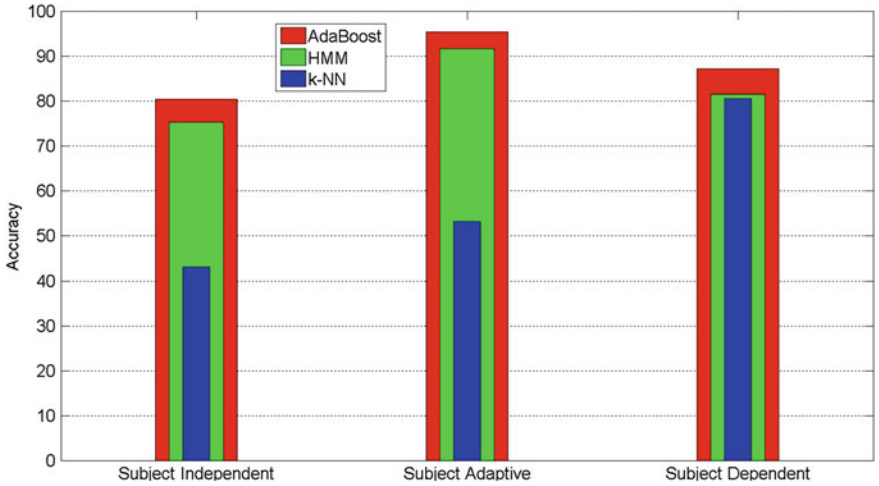


Fig. 14 Subject independent, adaptive, and dependent accuracies for AdaBoost, HMM and k-NN using the data from the accelerometer on right wrist

Table 7 Aggregate confusion matrix obtained for AdaBoost classification

Activity gesture	Lift to mouth	Pour	Scoop	Unscrew cap	Stir
Lift to mouth	95	0	5	0	0
Pour	3	94	3	0	0
Scoop	0	0	92	2	3
Unscrew cap	0	0	8	85	4
Stir	0	0	5	5	65

sive computation and so were not considered. Instead, we chose to use the fixed length features and a Euclidean distance based k-NN.

The subject independent, adaptive, and dependent accuracies obtained for gesture dataset are summarized in Fig. 14. Overall, recognition accuracy is highest for AdaBoost in the subject independent and adaptive evaluations with 90 and 95 % accuracy. HMMs were the second most accurate algorithm. The 14 lower recognition accuracy for AdaBoost for the subject dependent evaluation can be attributed to the reduction in the number of training samples. K-NN resulted in a mere 43 % subject independent accuracy.

Table 7 shows the aggregate confusion matrix for AdaBoost classifier based on the leave one out subject independent evaluation. Recognition accuracies for the stir action were 68 %, while lift to mouth, pour, and scoop had significantly better accuracies of 95, 94, and 92 %, respectively. There was a notable confusion between the actions stir and unscrew cap, both of which involve repetitive patterns with only marginal differences in the rate at which the action is performed. Thus, there is a chance of miss-classification when stir is performed quickly or unscrew cap is performed slowly.

Overall, the recognition accuracy was significantly higher for AdaBoost, probably indicating the need for a discriminative classifier that models the boundaries between the classes for recognition. The high accuracy obtained for the leave one out subject independent evaluation process indicates the presence of strong correlations in the action patterns of different subjects. In addition, since this evaluation used a large training data set (from 4 subjects), the training process would have resulted in a more generalized classifier.

5.4.1 Accelerometer Configuration

It is evident from the analysis of the ambulatory movements that placement of the accelerometers plays a crucial role in determining the performance of the system. This holds true for hand movements as well. The discriminatory power of the different accelerometer locations on the hand was evaluated using the AdaBoost classifier. As mentioned before, three locations for placing the accelerometers: right wrist, right elbow, and left wrist were used. Since all the subjects were right-handed, the right wrist is the most intuitive location to place the accelerometer. While the palm would be the position that can discriminate the most between the different actions, placing the accelerometer on the palm would restrict the individual's movement and hence was not used.

Table 8 summarizes the recognition accuracies obtained using data from these three accelerometer positions. The best performance was achieved with a combination of accelerometers on the right wrist and the right elbow. This is intuitive, as the subjects used their right hand for all the actions, resulting in a strong correlation between the motion at the wrist and the elbow. Thus monitoring the joints that determine the movement of the hands is useful for recognizing the actions involved with mixing and drinking a powdered beverage. However, movements from the right elbow alone are not sufficient to classify the actions. While our initial hypothesis was that data from the accelerometer on the left wrist, would aid in classification, surprisingly adding acceleration information from the left wrist reduced the performance. A review of the video recordings highlighted one possible explanation for this result: not all subjects used their left hand, nor did subjects use their left hands in a similar manner during the recorded activities.

Table 8 Classification accuracies for different configurations of the accelerometers

Accelerometer configuration	Classification accuracy
Right wrist	89.31
Right elbow	74.90
Right elbow and wrist	90.01
Right and left wrist	65.45

Table 9 Isolated recognition performance on CS2 dataset through classifiers trained on CS1, CS2, and a combination both the datasets

Classifiers	Gesture data from only CS1	Gesture data from only CS2	Combination of CS1 and CS2 datasets
AdaBoost	0.74	0.68	0.86

5.4.2 Performance Semi-Naturalistic Scenario Activity Gesture Dataset

The same experiment was repeated with the data collected in CS2 corresponding to the real-life scenario. Three training strategies were used to evaluate the performance in this scenario. In the first strategy, models learnt from data collected using CS1 were used to test samples from CS2. In the second strategy, samples from two trials of every subject obtained from CS2 were used for training. However, since the number of training samples available from data collected under CS2 is very less, samples collected from CS1 were also included in training in the third strategy. For training scenarios that used data from CS2, the trained models were tested on the remaining trials of the subject. The two test trials was chosen in a round robin fashion, for determining the most generalized accuracy in the given situation. The isolated recognition accuracies for each of the models are shown in Table 9. As expected, there is a drop in performance of AdaBoost when trained on samples using CS1. Since there is an inherent difference in the movement patterns between the scenarios as CS2 corresponds to an unconstrained environment. The accuracies obtained using models trained on data collected in real-life scenarios are also poor. This is probably because of the lack of sufficient number of training samples. Models were trained with approximately 10–20 samples per class. However, when the samples from both CS1 and CS2 were combined, the performance of AdaBoost improved drastically clearly indicating some similarity in the movement samples from both the scenarios that can assist in the learning process.

6 Conclusion

This chapter introduces an application of Body Sensor Networks that deals with recognizing simple ambulatory activities from wearable accelerometer data. It presents the generic framework for processing sensor data to recognize activities. It evaluates the effectiveness of different discriminative classifiers for simple activity recognition from low-resolution accelerometer data. A technique for post classification label smoothing based on the temporal continuity of the data was also discussed. This proposed technique for adding temporal continuity to the classification yielded promising results with about 2.5–3 % improvement in accuracy. It also discusses a methodology for taking into consideration human factors for improving the classification performance in the continuous scenario. Based on these results, a real-time

simple activity recognition system was designed and implemented. The proposed real-time system is able to accurately recognize the simple activities in real-time achieving a performance superior to existing approaches in the literature. The complexity of the system was discussed both in terms of the computational complexity and normalized time for computation for classifying a single sample.

This chapter also discussed the application of BSN for recognizing activity gestures. These gestures symbolize different spatio-temporal variations and pose a significant problem for automatic recognition. The performance of different discriminative, generative-, and template-based classification approaches for recognizing these gestures was studied on the gesture datasets collected from wearable accelerometers. The results indicate a superior performance of discriminative classifiers trained on statistical features extracted from the accelerometer data over generative and template-based approaches. This establishes the feasibility of accelerometer-based activity gesture recognition.

There have been a few recent studies similar to the on discussed in this chapter that use commercially available mobile devices to collect data for activity recognition. Kwapisz et al. [39] use an Android-based smart phone for recognizing simple activities, while Yang et. al [40] and Brezmes et. al [41] employ a Nokia N95 device for achieving the same goal. Hache et. al [42] use an accelerometer integrated in a Blackberry Bold 900 for detecting changes in the state of the subject. In a more recent work, Dernbach et. al [43] explore the ability of a Samsung Captivate smart phone for recognizing both simple and complex activities of daily living. Even with the rapid advances made by in the field of body sensor networks, there still remain many computational research challenges that have to be solved for integrating the BSN applications into the real world. These challenges include, improving the classification performance on data being sampled at a significantly lower sample rate, reducing the power consumed by the sensor, building on-chip computational algorithms for processing the data for reducing the data transmission from the sensors and handling the variations resulting from the wearability aspect of these sensors such as sensor slippage [76], orientation, and location changes. Current research in this area is focused on addressing some of these challenges.

Acknowledgments The authors would like to thank Colin Juillard, Dirk Colbry, Ashok Venkatesan, and Rita Chattopadhyay for the assistance they rendered in designing the real-time system and for collecting activity data from different subjects. They also would like to thank Stephen Intille for sharing the accelerometer-based activity data collected by his group which was used to conduct some of the experiments in this chapter.

References

1. F. Foerster, S. Smeja, J. Fahrenberg, Detection of posture and motion by accelerometery: a validation study in ambulatory monitoring. J. Comput. Hum. Behav. **15**, 571–583 (1999)
2. M. Uiterwaal, E.B.C. Glerum, H.J. Busser, R.C. van Lummel, Ambulatory monitoring of physical activity in working situations, a validation study. J. Med. Eng. Technol. **22**(4), 168–172 (1998)

3. M.J. Mathie, A.C.F. Coster, N.H. Lovell, B.G. Celler, Detection of daily physical activities using a triaxial accelerometer. J. Med. Biol. Eng. Comput. **41**, 296–301 (2003)
4. M.J. Mathie, A.C.F. Coster, N.H. Lovell, B.G. Celler, S.R. Lord, A. Tiedmann, A pilot study of long-term monitoring of human movements in the home using accelerometry. J. Telemed. Telecare **10**(3), 144–151 (2004)
5. R.K. Ibrahim, E. Ambikairajah, B.G. Celler, N.H. Lovell, Time-frequency based features for classification of walking patterns, in *Proceedings of International Conference on Digital Signal Processing*, pp. 187–190 (2007)
6. N.C. Krishnan, S. Panchanthan, Analysis of low resolution accelerometer data for continuous human activity recognition, in *Proceedings of International Conference on Acoustics, Speech and Signal Processing*, pp. 3337–3340 (2008)
7. J. Lester, T. Choudhury, N. Kern, G. Borriello, B. Hannaford, A hybrid discriminative/generative approach for modeling human activities, in *Proceedings of International Joint Conference on Artificial Intelligence*, pp. 766–772 (2005)
8. J. Suutala, S. Pirtikangas, J. Roning, Discriminative temporal smoothing for activity recognition from wearable sensors, in *Proceedings of International Symposium on Ubiquitous Computing Systems*, 2007), pp. 182–195
9. N. Ravi, N. Dandekar, M. Preetham, M.L. Littman, Activity recognition from accelerometer data, in *Proceedings of International Conference on Innovative Applications of Artificial Intelligence, IAAI*, pp. 1541–1546 (2005)
10. N. Bidargaddi, A. Sarela, L. Klingbeil, M. Karunanithi, Detecting walking activity in cardiac rehabilitation by using accelerometer, in *Proceedings of International Conference on Intelligent Sensors, Sensor Networks and Information*, pp. 555–560 (2007)
11. N.C. Krishnan, D. Colbry, C. Juillard, S. Panchanathan, Real time human activity recognition using tri-axial accelerometers, in *Proceedings of Sensors Signals and Information Processing Workshop* (2008)
12. D.M. Karantonis, M.R. Narayanan, M.J. Mathie, N.J. Lovell, B.G. Celler, Implementation of a real-time human movement classifier using a tri-axial accelerometer for ambulatory monitoring. IEEE Trans. Inf. Technol. Biomed. **10**(1), 156–167 (2006)
13. L. Bao, S.S. Intille, Activity recognition from user-annotated acceleration data, in *Proceedings of International Conference on Pervasive Computing*, vol. 3001, pp. 1–17 (2004)
14. R. Muscillo, S. Conforto, M. Schmid, P. Caselli, T. D'Alessio, Classification of motor activities through derivative dynamic time warping applied on accelerometer data, in *Proceedings of International Conference on Engineering in Medicine and Biology*, pp. 4930–4933 (2007)
15. J.A. Ward, P. Lukowicz, G. Troster, T.E. Starner, Activity recognition of assembly tasks using body worn microphones and accelerometers. IEEE Trans. Pattern Anal. Mach. Intell. **28**(10), 1553–1567 (2006)
16. H. Junker, O. Amft, P. Lukowicz, G. Toster, Gesture spotting with body worn inertial sensors to detect user activities. Pattern Recogn. **41**(6), 2010–2024 (2008)
17. J.F. Knight, H.W. Bristow, S. Anastopoulou, C. Baber, A. Schwirtz, T.N. Arvantis, Uses of accelerometer data collected from a wearable system. J. Pers. Ubiquit. Comput. **11**(2), 117–132 (2007)
18. T. Huynh, B. Schiele, Analyzing features for activity recognition, in *Proceedings of Joint Conference on Smart Objects and Ambient Intelligence: Innovative Context-Aware Services: Usages and Technologies*, pp. 159–163 (2005)
19. T. Al-ani, Q. Trang Le Ba, E. Monacelli, On-line automatic detection of human activity in home using wavelet and hidden markov models scilab toolkits, in *Proceedings of International Conference on Control Applications*, pp. 485–490 (2007)
20. K.V. Laerhoven, O. Cakmakci, What shall we teach our pants, in *Proceedings of International Symposium on Wearable Computers* (2000)
21. T. Huynh, B. Schiele, Unsupervised discovery of structure in activity data using multiple eigenspaces, in *Proceedings of International Workshop on Location and Context Awareness***3987**, 151–167 (2006)
22. http://www.sparkfun.com/commerce/categories.php

23. http://www.freescale.com/webapp/sps/site/taxonomy.jsp?code=ACCLOWG&tid=vanxyz
24. http://fiji.eecs.harvard.edu/Mercury
25. http://wockets.wikispaces.com/Wockets+at+Stanford
26. E. Munguia Tapia, S.S. Intille, L. Lopez, K. Larson, The design of a portable kit of wireless sensors for naturalistic data collection, in *Proceedings of PERVASIVE 2006* (Springer, Heidelberg, 2006)
27. R. Jafari, W. Li, W. Bajcsy, S. Glaser, S. Sastry, Physical activity monitoring for assisted living at home, in *Proceedings of International Workshop on Wearable and Implantable Body Sensor Networks***13**, 213–219 (2007)
28. N. Kern, B. Schiele, A. Schmidt, Multi-sensor activity context detection for wearable computing, in *Proceedings of European Symposium on Ambient Intelligence*, pp. 220–232 (2003)
29. F.R. Allen, E. Ambikairajah, N.H. Lovell, B.G. Celler, An adapted Gaussian mixture model approach to accelerometery based movement classification using time-domain features, in *Proceedings of International Conference on Engineering in Medicine and Biology Society* (2006)
30. D. Olgun, A. Pentland, Human activity recognition: accuracy across common locations for wearable sensors, in *Proceedings of International Symposium on Wearable Computing* (2006)
31. C.C. Tseng, D. Cook, Mining from time series human movement data, in *Proceedings of International Conference on Systems, Man, and Cybernetics*, pp. 3241–3243 (2006)
32. H.M. Yang, N. Ahuja, M. Tabb, Extraction of 2D motion trajectories and its application to hand gesture recognition. IEEE Trans. Pattern Anal. Mach. Intell. **24**, 1061–1074 (2002)
33. V.M. Mantyla, J. Mantyjarvi, T. Seppanen, E. Tuulari, Hand gesture recognition of a mobile device user, in *2000 IEEE International Conference on Multimedia and Expo, 2000, ICME 2000***1**, 281–284 (2000)
34. R.O. Duda, P.E. Hart, D.G. Stork, *Pattern Classification* (John Wiley & Sons, New York, 2001), pp. xx + 654, ISBN: 0-471-05669-3
35. T. Joachims, in *Making large-Scale SVM Learning Practical*, ed. by B. Schölkopf, C. Burges, A. Smola. Advances in Kernel Methods—Support Vector Learning (MIT-Press, Cambridge, 1999)
36. P. Komarek, A. Moore, Fast Robust Logistic Regression for Large Sparse Datasets with Binary Outputs (2003)
37. Y.A. Ivanov, A.F. Bobick, Recognition of visual activities and interactions by stochastic parsing. Trans. Pattern Anal. Mach. Intell. **22**(8), 852–872 (2000)
38. J. Boyd, H. Sundaram, A. Shrivastava, Power-accuracy tradeoffs in human activity transition detection. DATE 1524–1529 (2010)
39. J.R. Kwapisz ,M.W. Weiss, S.A. Moore, Activity recognition using cell phone accelerometers, in *Proceedings of the Fourth International Workshop on Knowledge Discovery from Sensor Data*, Washington DC, pp 10–18 (2010)
40. J. Yang, Toward physical activity diary: Motion recognition using simple acceleration features with mobile phones, in *First International Workshop on Interactive Multimedia for Consumer Electronics at ACM Multimedia* (2009)
41. T. Brezmes, J.L. Gorricho, J. Cotrina, Activity Recognition from accelerometer data on mobile phones, in *IWANN Proceedings of the 10th International Conference on Artificial Neural Networks*, pp. 796–799 (2009)
42. G. Hache, E.D. Lemaire, N. Baddour, Mobility change-of-state detection using a smartphone-based approach, in *Proceedings of International Workshop on Medical Measurements and Applications*, pp. 43–46 (2010)
43. S. Dernbach, B. Das, N.C. Krishnan, B.L. Thomas, D. Cook, Activity recognition on smart phones, in *IEEE International Conference on Intelligent Environments* (2012)
44. U. Maurer, A. Smailagic, D.P. Siewiorek, M. Deisher, Activity recognition and monitoring using multiple sensors on different body positions, in *Proceedings of the International Workshop on Wearable and Implantable Body Sensor Networks*, pp. 113–116 (2006)
45. S. Hiroaki, Dynamic programming algorithm optimization for spoken word recognition. IEEE Trans. Acoust. Speech Signal Processing **26**, 43–49 (1978)

46. A. Corradini, Dynamic time warping for off-line recognition of a small gesture vocabulary, in *Proceedings of the IEEE ICCV Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems*, p. 82 (2001)
47. T. Darrell, A. Pentland, Space-time gestures, in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 335–340 (1993)
48. S.A. Niyogi, E.H. Adelson, Analyzing and recognizing walking figures in xyt, in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 469–474 (1994)
49. D.M. Gavrila, L.S. Davis, Towards 3-d model-based tracking and recognition of human movement: a multi-view approach, in *International Workshop on Automatic Face- and Gesture-Recognition*. IEEE Computer Society, pp. 272–277 (1995)
50. A. Veeraraghavan, R. Chellappa, A.K. Roy-Chowdhury, The function space of an activity, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 959–968 (2006)
51. A. Jonathan, A unified framework for gesture recognition and spatiotemporal gesture segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **31**, 1685–1699 (2008)
52. T. Starner, A. Pentland, W. Joshua, Real-time american sign language recognition using desk wearable computer based video. IEEE Trans. Pattern Anal. Mach. Intell. **20**, 1371–1375 (1998)
53. C. Bregler, Learning and recognizing human dynamics in video sequences, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, p. 568 (1997)
54. K. Oka, Y. Sato, H. Koike, Real-time fingertip tracking. IEEE Gesture Recogn. **22**, 64–71 (2002)
55. T. Starner, J. Auxier, D. Ashbrook, M. Gandy, The gesture pendant: a self-illuminating, wearable, infrared computer vision system for home automation control and medical monitoring, in *Fourth International Symposium on Wearable Computers*, pp. 87–94 (2000)
56. M. Brand, Shadow puppetry, in *Proceedings of the International Conference on Computer Vision***2**, 1237 (1999)
57. Z. Ghahramani, Learning dynamic Bayesian networks. Lect. Notes Comput. Sci. **1387**, 168 (1998)
58. S. Park, J.K. Aggarwal, Recognition of two-person interactions using a hierarchical bayesian network, *ACM SIGMM International Workshop on Video Surveillance*, pp. 65–76 (2003)
59. P. Peursum, G. West, S. Venkatesh, Combining image regions and human activity for indirect object recognition in indoor wide-angle views. Proc. IEEE Int. Conf. Comput. Vis. **1**, 82–89 (2005)
60. N.T. Nguyen, D.Q. Phung, S. Venkatesh, H. Bui, Learning and detecting activities from movement trajectories using the hierarchical hidden markov models. Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recogn. **2**, 955–960 (2005)
61. N. Jojic, N. Petrovic, B.J. Frey, T.S. Huang, Transformed hidden markov models: estimating mixture models of images and inferring spatial transformations in video sequences, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 26–33 (2000)
62. K. Toyama, A. Blake, Probabilistic Tracking, with Exemplars in a Metric Space. Int. J. Computer Vision, Issue 1. Vol. 48, 9–19 (2002)
63. Y. Cui, J. Weng, Appearance-based hand sign recognition from intensity image sequences. Comput. Vis. Image Underst. **78**(2), 157–176 (2000)
64. E. Ong, R. Bowden, A boosted classifier tree for hand shape detection, *IEEE International Conference on Automatic Face and Gesture Recognition*, p. 889 (2004)
65. S. Wang, W. Pentney, A.M. Popescu, T. Choudhury, M. Philipose, Common sense based joint training of human activity recognizers, in *Proceedings of the International joint conference on Artificial Intelligence*, pp. 2237–2242 (2007)
66. H. Yang, S. Sclaroff, S. Lee, Sign language spotting with a threshold model based on conditional random fields. IEEE Trans. Pattern Anal. Mach. Intell. **31**, 1264–1277 (2008)
67. L.P. Morency, T. Darrell, Head gesture recognition in intelligent interfaces: the role of context in improving recognition, in *Proceedings of the International Conference on Intelligent User Interfaces*, pp. 32–38 (2006)

68. J.D. Lafferty, A. McCallum, F.C.N. Pereira, Conditional random fields: probabilistic models for segmenting and labeling sequence data, in *International Conference on Machine Learning*, pp. 282–289 (2001)
69. S. Kumar, M. Hebert, Discriminative random fields: a discriminative framework for contextual interaction in classification, in *International Conference on Computer Vision*, pp. 1150–1157 (2003)
70. C. Sminchisescu, A. Kanaujia, D. Metaxas, Conditional models for contextual human motion recognition. Comput. Vis. Image Underst. **104**(2), 210–220 (2006)
71. S.B. Wang, A. Quattoni, L.P. Morency, D. Demirdjian, Hidden conditional random fields for gesture recognition, in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1521–1527 (2006)
72. L.P. Morency, A. Quattoni, T. Darrell, Latent-dynamic discriminative models for continuous gesture recognition, in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1–8 (2007)
73. Y. Freund, R.E. Schapire, A short introduction to boosting. J. Jpn. Soc. Artif. Intell. **14**(5), 771–780 (1999)
74. Bishop C, Pattern Recognition and Machine Learning, Springer 2006.
75. J. Mannil, M. Bidmeshki, R. Jafari, Rejection of irrelevant human actions in real-time hidden Markov model based recognition systems for wearable computers, *ACM International Conference on Wireless Health*, pp. 10–13, October 2011
76. N. Kale, J. Lee, R. Lotfian, R. Jafari, Impact of sensor misplacement on dynamic, time warping based human activity recognition using wearable computers, *ACM International Conference on Wireless Health*, pp. 23–25, October 2012

Part VIII
Social Sensing

Chapter 16
Analytic Challenges in Social Sensing

Tarek Abdelzaher and Dong Wang

Abstract Social sensing applications refer to those where individuals play an important role in data collection. They can act as sensor carriers (e.g., carrying GPS devices that share location data), sensor operators (e.g., taking pictures with smart phones), or as sensors themselves (e.g., sharing their observations on Twitter). The proliferation of sensors in the possession of the average individual, together with the popularity of social networks that allow massive information dissemination, heralds an era of social sensing that brings about new research challenges reviewed in this chapter.

1 An Introduction to Social Sensing

The idea that individuals will play an important part in data collection pipelines has been around for some time in sensor network literature. Several surveys appeared in recent literature that offer different visions for the social sensing landscape [2, 5]. Pioneering work at UCLA defined *participatory sensing* as sensing that requires active participation from a human operator. Geo-tagging applications, where individuals need to explicitly mark locations of interest to the sensing application are a good representative of participatory sensing. Participatory sensing is often distinguished from *opportunistic sensing*, where individuals offer their sensing devices for transparent opportunistic exploitation by a distributed application for purposes of collecting data for the community. Finally, social networks, such as Twitter, offer new opportunities for utilizing human observations that are voluntarily reported on the social medium as another sensing modality of events in the physical world. This chapter collectively refers to the above sensing applications as *social sensing*. Hence, social sensing is defined as applications, where

T. Abdelzaher (✉) · D. Wang
University of Illinois Urbana-Champaign, Urbana, USA
e-mail: zaher@cs.uiuc.edu
e-mail: dwang24@illinois.edu

H. M. Ammari (ed.), *The Art of Wireless Sensor Networks*,
Signals and Communication Technology, DOI: 10.1007/978-3-642-40066-7_16,

humans play a key role in the data collection system by acting as sensor carriers (e.g., opportunistic sensing), sensor operators (e.g., participatory sensing), or sensors themselves. A confluence of *three social and technology trends* suggests that social sensing applications will have an increasingly important role in the future:

- *Sensing device proliferation:* The first trend that fuels social sensing applications is the commercial proliferation of sensing systems that are commonly accessible to large consumer populations. Active RFIDs, smart residential power meters (with a wireless interface), camera cell-phones, in-vehicle GPS devices, accelerometer-enhanced entertainment platforms (e.g., Wii-fit), and activity monitoring sports-ware (e.g., the Nike+iPod system) have all reached mature market penetration, offering unprecedented opportunities for data collection.
- *Mobile connectivity:* The second trend lies in ubiquitous mobile Internet access available to sensing platforms on the move. This untethered connectivity allows events to be measured and reported in real time, anytime, anywhere. A clear example is the case of GPS measurements and pictures taken by cell-phones. Besides GPS and cameras, modern smart phones currently host myriads of other sensors as well, such as accelerometers, magnetometers, and gyroscopes, and offer 3G/4G, WiFi and Bluetooth network access, which enables sharing their data. Vehicular Internet access, is also becoming available, for example, in recent models of Chrysler and BMW, which enables applications that exploit network connectivity. The vehicular OBD-II interface is already being used by services such as OnStar for remote diagnostics. Other applications may perform traffic statistics, alert to nearby accidents, or detect emergency conditions. In medical spaces, significant investments have been made in sensor technology for longitudinal monitoring. Microsoft HealthVault is one example of a recent initiative to automate collection of and access to medical information. A significant number of vendors announced wearable health and biometric monitoring sensors that automatically upload user data to HealthVault. The proliferation of sensing devices with Internet connectivity that collect data in social spaces makes it feasible to build human-centric data collection and sharing applications that augment human capabilities or improve "situation awareness."
- *Social networks:* The existence of sensors and Internet connectivity, however, is not necessarily sufficient, by itself, to support social sensing in the mainstream. The third key trend that fuels social sensing is the increased popularity of mass information dissemination channels, afforded by social networks. Twitter, Flickr, Twitpic, YouTube, and other networks allow individuals to globally broadcast their observations. It is this global dissemination opportunity that makes it easy to build large-scale applications that utilize commonly-available sensors, upload data in real-time, and share the observations at scale.

The above opportunities have generated significant interest in the research community in building application prototypes that rely on observations made by humans or by sensors in their vicinity. In the rest of this chapter, we briefly present recent work related to social sensing, categorize social sensing applications, then identify main challenges they need to overcome.

2 The Multidisciplinary Roots of Social Sensing: A Survey

This section describes the different analystic foundations and avenues of work related to Social sensing [14, 60]. An early overview of social sensing applications is presented in [1]. Some early applications include CenWits [43], a participatory sensor network to search and rescue hikers in emergency situations, CarTel [46], a vehicular sensor network for traffic monitoring and mitigation, CabSense [72], an application that analyzes GPS data from NYC taxis and helps you find the best corner to catch a cab [76] and BikeNet [29], a bikers sensor network for sharing cycling-related data and mapping the cyclist experience. In recent years, social sensing applications in healthcare have also become very popular. Numerous medical devices have been built with embedded sensors that can be used to monitor the personal health of patients, or send alters to the clinic or through the patient's social network when something unexpected happens. Such social sensing can be used for activity recognition for emergency response [56], long-term prediction of diseases [38, 57, 58], and lifestyle changes that affect health [20, 33].

Recent work in social sensing focused on challenges such as preserving privacy of participants [10, 67], improving energy efficiency of sensing devices [63, 64] and building general models in sparse and multi-dimensional social sensing spaces [9, 79]. Examples include privacy-aware regression modeling, a data fusion technique that produces the same model as that computed from raw data by properly computing non-invertible aggregates of samples [10]. Authors in [67] gave special attention to preserving privacy over time-series data based on the observation that a sensor data stream typically comprises a correlated series of sampled data from some continuous physical phenomena. Acquisitional Context Engine (ACE) is a middleware that infers unknown human activity attributes from known ones by exploiting the observation that the values of various human context attributes are limited by physical constraints and hence are highly correlated [63]. E-Gesture is an energy efficient gesture recognition architecture that significantly reduces the energy consumption of mobile sensing devices while keeping the recognition accuracy acceptable [64]. The sparse regression cube is a modeling technique that combines estimation theory and data mining techniques to enable reliable modeling at multiple degrees of abstraction of sparse social sensing data [9]. A further improved model to consider the data collection cost was proposed in [79].

Social sensing is often organized as "sensing campaigns" where participants are recruited to contribute their personal measurements as part of a large-scale effort to collect data about a population or a geographical area. Examples include documenting the quality of roads [71], measuring the level of pollution in a city [61], or reporting locations of garbage cans on campus [70]. In addition, social sensing covers scenarios where human sources spontaneously report data without prior coordination, such as data describing important events. Examples include large volumes of reported observations of political unrest, riots, and natural disasters on Twitter. The spread of social networks such as Twitter and You-tube offers a forum for global and real-time sharing of reported data, which makes the reporting especially powerful. This type of

applications represents a very broad, distributed and collaborative sensing paradigm that features the most versatile mobile platform, *the human user*, as the sensor. Recent research attempts to understand the fundamental factors that affect the behavior of these emerging social sensing applications, such as analysis of characteristics of social networks [23], information propagation [45] and tipping points [84].

A critical question about trustworthiness arises when the data in social sensing applications are collected by humans whose "reliability" is not known. In social sensing, anyone can contribute data. Such openness greatly increases the availability of the information and the diversity of its sources. On the other hand, it introduces the problem of understanding the reliability of the contributing sources and ensuring the quality of the information collected. Trusted Platform Module (TPM), commonly used in commodity PCs, can be used to provide a certain level of assurance at the expense of additional hardware [34]. YouProve is a recent technique that relies on trust analysis of derived data to allow untrusted client applications to verify that the meaning of source data is preserved [35]. Trust analysis can also be performed at the server side by building a likelihood function for sensed data to provide a quantifiable estimate of both source reliability and the correctness of observations.

2.1 Trust Analysis in Information Networks

To assess the credibility of facts reported from sources of unknown reliability, a relevant body of work in the machine learning and data mining communities performs trust analysis. Hubs and Authorities [53] used a basic fact-finder where the belief in a claim c is $B(c) = \sum_{s \in S_c} T(s)$ and the trustworthiness of a source s is $T(s) = \sum_{c \in C_s} B(c)$, where S_c and C_s are the sources asserting a given claim and the claims asserted by a particular source, respectively. Pasternack et al. extended the fact-finder framework by incorporating prior knowledge into the analysis and proposed several extended algorithms, such as *Average.Log, Investment, and Pooled Investment* [65]. Yin et al. introduced *TruthFinder* as an unsupervised fact-finder for trust analysis that works on a providers-facts network [86]. Other fact-finders enhanced the basic framework by incorporating analysis of properties or dependencies within claims or sources. Galland et al. [31] took the notion of hardness of facts into consideration. The source dependency detection problem was discussed and several solutions were proposed [12, 25, 26]. More recent work adapted Bayesian analysis to model source trustworthiness in an explicit and probabilistic way and improved the accuracy of truth estimation. Wang et al. [78] proposed the Bayesian Interpretation scheme as an approximation approach to correctly quantify the likelihood of correctness of conclusions obtained from the basic fact-finding scheme. Zhao et al. [90] presented another Bayesian approach to model different types of errors made by sources and merge multi-valued attribute types of entities in data integration systems. Additionally, trust analysis was done both on a homogeneous network [11, 87] and a heterogeneous network [74]. Fact-finding in the case of social sensing is more challenging due to the highly dynamic nature of social sensing applications [6].

The outputs of fact-finders are, in general, *rankings* of credibility values of sources and facts, which cannot be used to directly *quantify* participant reliability or measurement correctness in social sensing. Recent work established the relation between ranking outputs of fact-finders and posterior probabilities of participant reliability and measurement correctness by using Bayesian analysis. A maximum likelihood estimator, based the Expectation Maximization (EM) scheme, will be discussed later in this chapter.

There exists a good amount of literature in the machine learning community to improve data quality and identify low quality labelers in a multi-labeler environment. Sheng et al. proposed a repeated labeling scheme to improve label quality by selectively acquiring multiple labels and empirically comparing several models that aggregate responses from multiple labelers [73]. Dekel et al. applied a classification technique to simulate aggregate labels and prune low-quality labelers in a crowd to improve the label quality of the training dataset [22]. While applicable to social sensing, some of the above approaches make explicit or implicit assumptions that may be limiting. For example, the work in [73] assumed labelers were known a priori and could be explicitly asked to label certain data points. The work in [22] assumed most of labelers were reliable and the simple aggregation of their labels would be enough to approximate the ground-truth. In general, participants in social sensing upload their measurements at will, based on their own preferences, and the simple aggregation technique (e.g., majority voting) was shown to be inaccurate when the reliability of participants is not sufficient [65]. We later describe a recent maximum likelihood estimation approach that intelligently casts the QoI quantification problem in social sensing into an optimization problem that can be efficiently solved by an EM scheme.

2.2 Estimation Theory

In estimation theory, Expectation Maximization (EM) is a general optimization technique for finding the maximum likelihood estimation of parameters in a statistic model where the data are "incomplete" or involve latent variables in addition to estimation parameter and observed data [24]. That is, either there are some missing value among the data, or the model can be formulated more simply by assuming the existence of some unobserved data. The general EM algorithm iterates between two main steps: the Expectation step (E-step) and the Maximization step (M-step) until the estimation converges (i.e., the likelihood function reaches the maximum). In the E-step, the algorithm computes the expectation of the log-likelihood function (so-called Q-function) of complete data with respect to the conditional distribution of the latent variables given the current settings of the parameters and the observed data. In the M-step, it re-estimates the parameters in the next iteration that maximizes the expectation of the log-likelihood function defined in the E-step. EM is frequently used for data clustering in data mining and machine learning. For language modeling, the EM is often used to estimate parameters of a mixed model where the exact model

from which the data is generated is unobservable [89]. There are many good tutorials on EM algorithms [13, 59]. In recent work, it was shown that social sensing applications lend themselves nicely to an EM formulation. The optimal solution, in the sense of maximum likelihood estimation, directly leads to an accurate quantification of measurement correctness as well as participant reliability.

The Cramer-Rao lower bound (CRLB) is a fundamental bound used in estimation theory to characterize the lower bound on the estimation variance of a deterministic parameter [21]. The bound states that the variance of any unbiased estimator is lower-bounded by the inverse of the Fisher information. The partial derivative with respect to the estimation parameter of the log-likelihood function is called the score. The Fisher information is defined as the second moment of the score vector of random variable and estimation parameter [40]. The Fisher information is a way of measuring the amount of information that an observable random variable X carries about an unknown estimation parameter θ upon which the probability of X depends. Intuitively, if the Fisher information is large, the distribution with the θ_0 (i.e., true value) of the estimation parameter will be different and well distinguished from the distributions with parameter that is not so close to θ_0. This means we are able to estimate θ_0 well (hence a small variance) based on the data. Conversely, if the Fisher information is small, our estimation will be worse. One of the key properties of maximum likelihood estimation (MLE) is asymptotic normality. This property basically states that the MLE estimator is asymptotically distributed with a normal distribution as the data sample size goes up [15]. The mean of the normal distribution is the MLE of the estimation parameter and the variance is given by the CRLB of the estimation. Recent work derived an approach to compute the *confidence interval* in participant reliability estimation based on both the real and asymptotic CRLB by leveraging the asymptotic normality of the MLE estimator.

2.3 Outlier Analysis and Attack Detection

Several previous efforts on data cleaning and outlier analysis from data mining and noise removal from statistics addressed some notion of noisy data [27, 28, 47–49, 51]. They differ in the assumption made, the modeling approach applied and the objective targeted. For example, Bayesian inference and decision tree induction techniques are applied to fill the missing values of data by predictions from their constructed model [28]. Binning and linear regression techniques are used to smooth the noisy data by either using bin means or fitting data into some linear functions [47, 49]. Clustering techniques are widely used to detect outliers by organizing similar data values into clusters and identifying the ones that fall outside the clusters as outliers [48]. Other approaches are used in statistics to filter noise from continuous data [27, 51]. The Kalman filter is an efficient recursive filter that estimates some latent variables of a linear dynamic system from a series of noisy measurements [51]. It produces estimates of the measurements by computing a weighted average of the predicted values based on their uncertainty. Particle filters are more sophisticated

filters that are based on Sequential Monte Carlo methods. They are often used to determine the distribution of a latent variable whose state space is not restricted to a Gaussian distribution [27]. The above techniques will likely improve the quality of data analysis in social sensing.

In intrusion detection, one critical task is to detect (or identify) malicious nodes (or sources) accurately and confidently. Two main kinds of detection techniques exist: signature-based detection and anomaly-based detection [48, 82]. The signature-based detection takes the predefined attack patterns (by domain experts) as signatures and monitors the node's behavior (or network traffic) for matches to report the anomaly [48]. The anomaly-based detection builds profiles of normal node's (or network) behavior and use the profiles to detect new patterns that have a remarkable deviation [82]. For the QoI quantification problem in social sensing, identifying malicious patterns can help estimation participant reliability.

Since people are an indispensable element in social sensing, it is important to address "bad" sources and defend against a series of common attacks. The collusion attack is a common attack carried out by a group of individuals who collectively perform some malicious (sometimes illegal) actions based on their agreement to defraud honest sources or obtain an unfair advantage in the system. This attack could be mitigated by monitoring the interactions or relationships among attackers or identifying abnormal behavior from the group [55]. The sybil attack is another important attack carried out by a single attacker who intentionally creates a large number of independently named entities and uses them to gain a disproportionately large influence within the system. This attack could be mitigated by increasing the cost of creating identities and limiting the resources the attacker can use to create new identities [88].

2.4 Recommender and Reputation Systems

Social sensing is related to a type of information filtering systems, called recommender systems, where the goal is usually to predict a user's rating or preference for an item using a model built from the characteristics of the item and the behavioral pattern of the user [4]. Maximum-likelihood estimation has been used in collaborative recommender systems as a clustering module based on user interests [62], as well as in content-based recommender systems as a weighting factor estimator [68]. Recommender systems can help filter results of social sensing applications. For example, by offering the user knobs that decide trust weights in sources, a recommender system can help a user zoom in on trusted measurements. Consider a user who identifies several facts as assuracte in results of participatory sensing. The recommender system may then give several additional recommendations on data to look at based on similarity in sources, observed locales, or observation conditions.

Social sensing is also related to reputation systems. The basic idea of reputation systems is to let entities rate each other (e.g., after a transaction) or review some objects of common interest (e.g., products or dealers). The aggregated ratings

or reviews can then be used to derive trust or reputation scores, which can help other entities in deciding whether or not to trust a given entity or purchase a certain object [50]. Different types of reputation systems are being used successfully in commercial online applications. For example, eBay is a type of reputation system based on homogeneous peer-to-peer systems, which allows peers to rate each other after each pair of them conduct a transaction [3, 41]. The Amazon online review system represents another type of reputation systems, where different sources offer reviews on products (or brands, companies) they have experienced. Customers are affected by those reviews (or reputation scores) in making purchase decisions. Writing a review or a reputation assessment can be thought of as an instance of social sensing, where humans seriver information on other humans or objects, much like a sensor would measure a variable in its external environment. Note that, reputation systems are in general vulnerable to several attacks such as self-promoting, slandering, and denial of service [39]. Many of these attacks actually originate from collusion and Sybil attacks that we mentioned earlier, making it important to address security issue in social sensing.

The above has been a broad overview of work related to social sensing in different communities. with this general background, next, we classify social sensing applications and detail selected research challenges that arise in their context.

3 A Functional Taxonomy of Social Sensing Applications

One can generally divide social sensing applications into three types, depending on their functionality. The three types differ in the level of complexity of data processing done to the measurements.

- *Data-point-centric applications:* The first and simplest social sensing application is one where individuals share single data points (the observations) that are then made available to clients or decision-makers. An example is geo-tagging applications, where individuals share pictures (tagged by location) of entities of relevance to the application. For example, a sensing campaign might ask participants to document locations of invasive spieces in a park, or locations of garbage on a beach. These observations (and pictures) can then be displayed on a map, or offered to municipal decision-makers for appropriate action.
- *Statistics-centric applications:* The second type of social sensing applications is one where statistics are computed from the data. An example might be a traffic speed monitoring or a pollution monitoring application where the speed or pollution levels measured by different individuals are used to compute statistics such as averages and probability distributions. Many early examples of social sensing belong in that category. For example, traffic patterns were monitored in a city to help drivers avoid congestion areas [46], bike route data were collected by biking enthusiasts to help them pick better routes [29], and hiker encounters were recorded on mountain trails to help locate missing hikers [43]. These applications

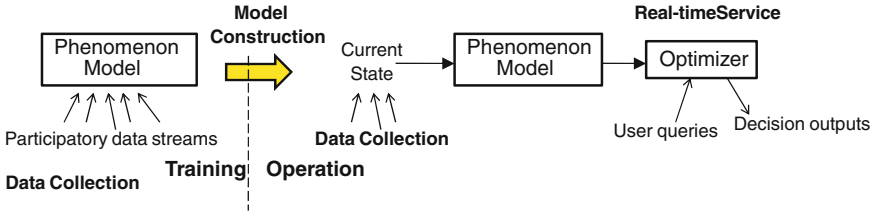


Fig. 1 An application template

offer useful statistics about a given locale, that are of interest to individuals in that locale.

- *Model-centric applications:* A third and most general type of social sensing applications has recently been described in literature, where *generalizable models* are learned from sensory data collection, that can be used to affect human decision making outside of the collection locale. For example, sharing data collected by smart energy meters installed in some households, together with relevant context, can lead to a better understanding of energy consumption in contemporary homes and best practices that increase energy efficiency elsewhere around the nation. Similarly, sharing data collected by activity sensors among fitness enthusiasts can lead to lifestyle recipes that promote healthier behaviors for multitudes of others. Also, sharing data on environmental pollutants and personal well-being (e.g., locations and incidents of asthma attacks) can establish links between likelihood of attacks and exposure to specific contaminants, which may help individuals reduce their exposure to those contaminants. For example, in a recent study of vehicular fuel-efficiency, a model predicts the total fuel consumption for a vehicle on a road segment as a function of several variables such as road speed, degree of congestion, and vehicle parameters. Once the model is known, it is possible to optimize human decisions by offering better (GPS) navigation advice for any vehicle on any street. Figure 1.

In the rest of this chapter, we cover three social sensing challenges, each motivated by one of the above application categories. These challenges are:

- *Challenge #1: Fact-finding.* Motivated by the need of data-point-centric applications, fact-finding addresses the problem of assertaining the correctness of reported observations. The individuals involved in data collection might report poor quality data, offer incorrect measurements, or inappropriately operate sensors. The reliability of such individuals and measurements may not, in general, be known to the collection point. It may further be expensive to verify the correctness of each observation. Hence, new techniques are needed to assess the credibility of data.
- *Challenge #2: Privacy-preserving data collection.* In observation-centric and statistics-centered applications, another issue that arises is the privacy of reporting individuals. Unless the application is managed by a globally trusted authority (i.e., one that is entrusted with private data), ensuring the privacy of data shared could

be a concern. Anonymity is not always a sufficient solution because the data themselves (such as GPS traces) may reveal the identity of the owner even if shared anonymously. One question becomes whether it is possible to perturb the data in a way that protects privacy, but without degrading application quality. For example, in statistics-centered applications, can one perturb the individual data points (for privacy reasons) without affecting the statistical properties of the aggregate?

- *Challenge #3: Generalized model construction.* In model-centric applications, models of the measured system might have a significant number of parameters, non-linearities, and discontinuities. Learning the model implies partitioning the input parameter space into subspaces within which individual models apply, and deriving the best model in each subspace. For example, in a vehicular fuel consumption modeling application, the most important predictors of fuel efficiency of cars may depend on the type of car, make, year, or other inputs. Understanding how best to generalize across different cars is not an easy undertaking. Some model inputs may be static (e.g., car weight). Some may be dynamic (e.g., traveled road speed and degree of congestion). In many cases, the space of possible parameters is very large. It is difficult to predict *a priori* which parameters will be more telling and for what subset of the input space. It is the responsibility of the social sensing service to offer a general mechanism for applications to partition the input data space appropriately and build good models quickly from the data collected. This is complicated by the high-dimensionality of the problem space and the relatively sparse sampling of that space by users. New learning techniques are needed to address this challenge.

The above challenges are described in the following sections, respectively.

4 Challenge #1: Fact-Finding

In this section, we establish the analytic foundation and general software framework for fact-finding (in the absence of prior information about the individual sources). Fact-finding, in data-point-centric applications, refers to the question of ascertaining the correctness of individual data points, and the reliability of the sources, given niether pieces of information in advance. Hence, underlying the fact-finding process is the abstraction of *sources* (e.g., sensors and people) and *claims* (the observations they make). The goal is simply to select (i.e., "distill") from the pool of all claims only those that exceed a certain credibility threshold.

Claims, in our framework, are a very general notion that can be applied in many different settings. The main requirement is that one can identify corroborating claims. Towards that end, we define a *distance function* that describes how similar or different two claims are. For example, if claims refer to sensory data, such as temperature values, the distance function could simply be the difference in temperature. If claims refer to pieces of text, a distance function might be the Jaccard distance [75], a commonly used metric for deciding how similar two pieces of text are. (More complex

functions are needed to identify topic similarity, and recognize synonyms, negation, etc.) Finally, if claims refer to images taken, a distance function might be the color correlogram [42] (or a more complex similarity metric). Note that, claims can also be multi-dimensional. For example, if temperature is sensed at different locations and different times of day, one can think of these measurements as points in a space whose dimensions are sensor value (temperature), location, and time of day. A distance metric such as an (appropriately weighted) L2 norm can be defined between points in that space. The distance function plays a very important role in that it enables *similar claims* to be clustered together. Examples of similar claims include tweets that say (approximately) the same thing, pictures of (approximately) the same scene, or similar sensory measurements from the same location and time. Such similar claims can be consolidated into one, thus forming a source-claim network, where (in general) sources make multiple claims and claims are corroborated by multiple sources.

The source-claim network, formed as described above, is a general representation of reported data that enables fact-finding. For simplicity of illustrating the fact-finding process, let us consider the case where claims are binary. In general, let S_iC_j denote an observation reported by source S_i claiming that C_j is true. For example, in a city cleaning effort, a source might report that location j has offensive wall graffiti. Let $P(C_j^t)$ and $P(C_j^f)$ denote the probability that the variable C_j is indeed true or false, respectively. Different sources may make different numbers of observations. Let us define a_i as the (unknown) probability that source S_i reports a measured variable to be true when it is indeed true, and b_i as the (unknown) probability that source S_i reports a measured variable to be true when it is in reality false. The only input to the algorithm is the source-claim matrix SC, where $S_iC_j = 1$ when source S_i reports that C_j is true, and $S_iC_j = 0$ otherwise.

4.1 Expectation Maximization

One way to assess the correctness of sources and claims is to use the Expectation-Maximization (EM) algorithm. EM is a general algorithm for finding the maximum likelihood estimates of parameters in a statistical model, where the data are "incomplete" or the likelihood function involves latent variables [24]. Intuitively, what EM does is iteratively "completes" the data by "guessing" the values of hidden variables then re-estimates the parameters by using the guessed values as true values.

The main challenge in using the EM algorithm lies in the mathematical formulation of the problem in a way that is amenable to an EM solution. Given an observed data set X, one should judiciously choose the set of latent or missing values Z, and a vector of unknown parameters θ, then formulate a likelihood function $L(\theta; X, Z) = p(X, Z|\theta)$. Once the formulation is complete, the EM algorithm iteratively finds the maximum likelihood estimate.

The social sensing problem fits nicely into the Expectation Maximization (EM) model. First, let us introduce a latent variable vector Z where $z_j = 1$ when C_j is true

and $z_j = 0$ otherwise. We further denote the source-claim matrix SC as the observed data X, and take $\theta = (a_1, a_2, ...a_M; b_1, b_2, ...b_M; d)$ as the parameter of the model that we want to estimate. The goal is to get the maximum likelihood estimate of θ for the model containing observed data X and latent variables Z. The likelihood function $L(\theta; X, Z)$ is given by:

$$\begin{aligned} L(\theta; X, Z) &= p(X, Z|\theta) \\ &= \prod_{j=1}^{N} \left\{ \prod_{i=1}^{M} a_i^{S_i C_j} (1 - a_i)^{(1 - S_i C_j)} \times d \times z_j \right. \\ &\quad \left. + \prod_{i=1}^{M} b_i^{S_i C_j} (1 - b_i)^{(1 - S_i C_j)} \times (1 - d) \times (1 - z_j) \right\} \end{aligned} \tag{1}$$

where, as we mentioned before, M and N refer to the number of sources and measured variables, respectively. Parameters a_i and b_i are the conditional probabilities that source S_i reports variable C_j to be true given that C_j is in fact true or false, respectively. $S_i C_j = 1$ when source S_i reports that C_j is true, and $S_i C_j = 0$ otherwise, and d is the background bias that a randomly chosen measured variable is true.

Given the above formulation, it is shown in recent work [81] that EM, applied to the above problem, results in two simple equations that can be solved iteratively to compute the conditional probability that the latent variable z_j (and hence claim C_j), as well as the reliability of sources, expressed by parameters a_i^*, b_i^* are computed for the current value of z_j. Specifically, the Expectation step (E-step) becomes:

$$\begin{aligned} Q\left(\theta|\theta^{(t)}\right) &= E_{Z|X,\theta^{(t)}}[\log L(\theta; X, Z)] \\ &= \sum_{j=1}^{N} \left\{ p(z_j = 1|X_j, \theta^{(t)}) \right. \\ &\quad \times \left[\sum_{i=1}^{M} (S_i C_j \log a_i + (1 - S_i C_j) \log(1 - a_i) + \log d) \right] \\ &\quad + p(z_j = 0|X_j, \theta^{(t)}) \\ &\quad \left. \times \left[\sum_{i=1}^{M} (S_i C_j \log b_i + (1 - S_i C_j) \log(1 - b_i) + \log(1 - d)) \right] \right\} \end{aligned} \tag{2}$$

where X_j represents the jth column of the observed SC matrix (i.e., observations of the jth measured variable from all participants) and $p(z_j = 1|X_j, \theta^{(t)})$ is the conditional probability of the latent variable z_j to be true given the observation matrix related to the jth measured variable and current estimate of θ, which is given by:

$$
\begin{aligned}
p(z_j = 1|X_j, \theta^{(t)}) &= \frac{p(z_j = 1; X_j, \theta^{(t)})}{p(X_j, \theta^{(t)})} \\
&= \frac{p(X_j, \theta^{(t)}|z_j = 1)p(z_j = 1)}{p(X_j, \theta^{(t)}|z_j = 1)p(z_j = 1) + p(X_j, \theta^{(t)}|z_j = 0)p(z_j = 0)} \\
&= \frac{A(t, j) \times d^{(t)}}{A(t, j) \times d^{(t)} + B(t, j) \times (1 - d^{(t)})}
\end{aligned} \tag{3}
$$

where $A(t, j)$ and $B(t, j)$ are defined as:

$$
\begin{aligned}
A(t, j) &= p(X_j, \theta^{(t)}|z_j = 1) \\
&= \prod_{i=1}^{M} a_i^{(t)S_iC_j}(1 - a_i^{(t)})^{(1-S_iC_j)} \\
B(t, j) &= p(X_j, \theta^{(t)}|z_j = 0) \\
&= \prod_{i=1}^{M} b_i^{(t)S_iC_j}(1 - b_i^{(t)})^{(1-S_iC_j)}
\end{aligned} \tag{4}
$$

$A(t, j)$ and $B(t, j)$ represent the conditional probability regarding observations about the jth measured variable and current estimation of the parameter θ given the jth measured variable is true or false respectively.

Next we simplify Eq. (2) by noting that the conditional probability of $p(z_j = 1|X_j, \theta^{(t)})$ given by Eq. (3) is only a function of t and j. Thus, we represent it by $Z(t, j)$. Similarly, $p(z_j = 0|X_j, \theta^{(t)})$ is simply:

$$
\begin{aligned}
p(z_j = 0|X_j, \theta^{(t)}) &= 1 - p(z_j = 1|X_j, \theta^{(t)}) \\
&= \frac{B(t, j) \times (1 - d^{(t)})}{A(t, j) \times d^{(t)} + B(t, j) \times (1 - d^{(t)})} \\
&= 1 - Z(t, j)
\end{aligned} \tag{5}
$$

Substituting from Eqs. (3) and (5) into Eq. (2), we get:

$$
\begin{aligned}
Q\left(\theta|\theta^{(t)}\right) = \sum_{j=1}^{N} \Bigg\{ & Z(t, j) \\
& \times \left[\sum_{i=1}^{M} (S_iC_j \log a_i + (1 - S_iC_j) \log(1 - a_i) + \log d) \right]
\end{aligned}
$$

$$+ (1 - Z(t, j)) \times \left[\sum_{i=1}^{M} (S_i C_j \log b_i + (1 - S_i C_j) \log(1 - b_i) + \log(1 - d)) \right] \Bigg\} \tag{6}$$

To compute the maximization step (M-step), we choose θ^* (i.e., $(a_1^*, a_2^*, \ldots a_M^*; b_1^*, b_2^*, \ldots b_M^*; d^*)$) that maximizes the $Q\left(\theta|\theta^{(t)}\right)$ function in each iteration to be the $\theta^{(t+1)}$ of the next iteration.

To get θ^* that maximizes $Q\left(\theta|\theta^{(t)}\right)$, we set the derivatives $\frac{\partial Q}{\partial a_i} = 0$, $\frac{\partial Q}{\partial b_i} = 0$, $\frac{\partial Q}{\partial d} = 0$ which yields:

$$\begin{aligned}
&\sum_{j=1}^{N} \left[Z(t, j)(S_i C_j \frac{1}{a_i^*} - (1 - S_i C_j) \frac{1}{1 - a_i^*}) \right] = 0 \\
&\sum_{j=1}^{N} \left[(1 - Z(t, j))(S_i C_j \frac{1}{b_i^*} - (1 - S_i C_j) \frac{1}{1 - b_i^*}) \right] = 0 \\
&\sum_{j=1}^{N} \left[Z(t, j) M \frac{1}{d^*} - (1 - Z(t, j)) M \frac{1}{1 - d^*}) \right] = 0
\end{aligned} \tag{7}$$

Let us define SJ_i as the set of measured variables the participant S_i actually observes in the observation matrix SC, and $\bar{SJ_i}$ as the set of measured variables participant S_i does not observe. Thus, Eq. (7) can be rewritten as:

$$\begin{aligned}
&\sum_{j \in SJ_i} Z(t, j) \frac{1}{a_i^*} - \sum_{j \in \bar{SJ_i}} Z(t, j) \frac{1}{1 - a_i^*} = 0 \\
&\sum_{j \in SJ_i} (1 - Z(t, j)) \frac{1}{b_i^*} - \sum_{j \in \bar{SJ_i}} (1 - Z(t, j)) \frac{1}{1 - b_i^*} = 0 \\
&\sum_{j=1}^{N} \left[Z(t, j) \frac{1}{d^*} - (1 - Z(t, j)) \frac{1}{1 - d^*}) \right] = 0
\end{aligned} \tag{8}$$

Solving the above equations, we can get expressions of the optimal a_i^*, b_i^* and d^*:

$$\begin{aligned}
a_i^{(t+1)} = a_i^* &= \frac{\sum_{j \in SJ_i} Z(t, j)}{\sum_{j=1}^{N} Z(t, j)} \\
b_i^{(t+1)} = b_i^* &= \frac{K_i - \sum_{j \in SJ_i} Z(t, j)}{N - \sum_{j=1}^{N} Z(t, j)}
\end{aligned}$$

$$d_i^{(t+1)} = d_i^* = \frac{\sum_{j=1}^{N} Z(t, j)}{N} \tag{9}$$

where K_i is the number of measured variables observed by participant S_i and N is the total number of measured variables in the observation matrix. $Z(t, j)$ is defined in Eq. (3).

Given the above, The E-step and M-step of EM optimization reduce to simply calculating Eqs. (3) and (9) iteratively until they converge. The convergence analysis has been done for the EM scheme in prior work [83].

4.2 Analysis of Confidence

Netx, it is possible to derive a confidence interval based on the Cramer-Rao Bound for the aforementioned formulation of maximum likelihood estimation of source reliability [80].

In statistical mathematics and information theory, the Fisher information is a way of measuring the amount of information that an observable random variable X carries about an estimated parameter θ upon which the probability of X depends. The partial derivative of the log-likelihood function with respect to θ is called the score vector. The Fisher information is defined as the second moment of the score vector. It takes the form of an $k \times k$ matrix, where k is the number of elements in θ. In estimation theory and statistics, the Cramer-Rao bound (CRB) expresses a lower bound on the variance of estimators of a deterministic parameter. In its simplest form, the bound states that the variance of any unbiased estimator is at least as high as the inverse of the Fisher information [40]. The estimator that reaches this lower bound is said to be *efficient*.

An unbiased maximum likelihood estimator has the property that estimation error covariance reaches the Cramer-Rao bound (i.e., it is an efficient estimator). Hence, we can use the Cramer-Rao bound to derive a confidence interval that quantifies the accuracy of the estimated parameters, θ. Computing the Fisher Information Matrix from the likelihood function described in Eq. 1 for the truth assignment in our source-claim network, and given the converged estimate of $\hat{\theta}_{MLE}$ from EM, the estimation error covariance matrix, $Cov(\hat{\theta}_{MLE})$, can be written as:

$$(Cov(\hat{\theta}_{MLE}))_{i,j} \tag{10}$$
$$= \begin{cases} 0 & i \neq j \\ \dfrac{1}{\frac{\sum_{j \in SJ_i} Z_j^c}{(\hat{a}_i^{MLE})^2} + \frac{\sum_{j \in S\bar{J}_i} Z_j^c}{(1-\hat{a}_i^{MLE})^2}} & i = j \in [1, M] \\ \dfrac{1}{\frac{K_i - \sum_{j \in SJ_i} Z_j^c}{(\hat{b}_i^{MLE})^2} + \frac{N - K_i - \sum_{j \in S\bar{J}_i} Z_j^c}{(1-\hat{b}_i^{MLE})^2}} & i = j \in (M, 2M] \end{cases}$$

The estimation error of each element in the estimation parameter θ (i.e., a_i and b_i) follows an asymptotic norm distribution respectively with 0 mean and variance given by the main diagonal elements of the covariance matrix specified by Eq. (10). From this variance, a confidence interval in the estimates can be readily computed.

4.3 Evaluation

An evaluation of the above maximum likelihood estimation algorithm and confidence bound shows that it is successful at estimating credibility and error bounds. To evaluate fact-finding in data-point-centric applications, participants were asked to report locations of free (i.e., non-pay) parking lots on campus. In the experiment, 30 participants were invited, and 106 parking lots were surveyed (46 of which were indeed free). There were 901 reports collected from participants, some with conflicting data. Analysis using the proposed algorithm to determine which lots are indeed free resulted in less than 10 % false positives/negatives, compared to more than 20 % when simple majority (vote) is used. A larger-scale experiment was performed with Twitter, collecting data on major international events, where each tweet was considered a "claim" and similar tweets were clustered together into one, thereby forming a source-claim network that conformed to our theoretical model. Millions of tweets were collected on current events. Top tweets were then computed and manually verified to be true (the individuals who verified them had access to material published later in newspapers and online sources). It was shown that all tweets deemed reliable by the fact-finder were verified to be true according to the news. The Twitter experiments are presented in prior work [80, 81].

The above theoretical results and their experimental validation allow us to (i) compute a maximum likelihood estimate in the correctness of different claims and sources given only the information on who said what, and (ii) quantify our statistical confidence in the correctness of that estimate. In particular, the estimate is derived with neither prior knowledge of the reliability of sources nor independent means of verifying the claims. It is this absence of requirements for prior knowledge that makes the above approach especially suitable for social sensing applications, where sources (e.g., individuals who downloaded the social sensing phone app) may not be vetted and where their claims may be hard to independently verify. The above demonstrates that reliable information can indeed be distilled with confidence from unreliable sources, which is a key challenge in social sensing.

5 Challenge #2: Privacy-Preserving Data Collection

Ensuring privacy in data sharing is a key challenge in promoting social sensing. Since the sensor data in many cases (e.g., in the case of sharing GPS traces) may reveal the owner's identity even if shared anonymously, solutions other than anonymity are

needed to protect privacy. We shall not consider solutions that require the application itself to be globally trusted with private data (i.e., trusted by all participants) as such a trust requirement would significantly increase barrier to entry, impeding the introduction of new applications. Instead, below, we describe the mathematical foundations for *perturbing* time-series sensor data at the source for purposes of privacy-preserving sharing without significantly impacting the accuracy of models derived from community data. We define user privacy as the degree of uncertainty or error incurred in estimating the user's private data given the shared (perturbed) data. Our challenge is therefore to perturb a user's sequence of shared data values prior to sharing, such that: (i) the individual user data and trends (i.e., data changes with time) cannot be estimated without large error, whereas (ii) the distribution of the data aggregation results from all users and any models derived from the data can still be estimated with high accuracy.

A significant amount of research was done on data perturbation in the database community in the context of privacy-preserving data mining. Our work is different in addressing the scenario of sensing applications, marked by (i) sharing of one or more, possibly correlated, *time-series* data streams, as opposed to individual values, (ii) a high-dimensional data space, and (iii) the desire to learn regression models from data. Privacy-preserving perturbation of correlated time-series data that allows reconstruction of accurate multidimensional statistics and regression models from perturbed data is an exciting new problem. Of particular interest is the derivation of mathematical *privacy measures* and *privacy bounds* on such measures that quantify the efficacy of a perturbation scheme at hiding private information.

Examples of data perturbation techniques can be found in [7, 8, 30]. Early approaches relied on adding independent random noise. These approaches were shown to be inadequate because independent noise is easy to filter out [52] when the underlying data are correlated. Later approaches took data correlation among different users into account [44]. However, they did not make assumptions on the model describing the *evolution* of data values obtained from the same user over time, which can be used to jeopardize privacy of time-series data streams.

5.1 A Privacy-Preserving Perturbation Algorithm

Recent work [32], presented a perturbation technique that preserves privacy of time-series data. It demonstrated that private time-series data could not be recovered from perturbed time-series data by eliminating added noise using common techniques such as spectral filtering [52] and principal component analysis (PCA) [44]. It also demonstrated client-side tools that automate noise generation for client data. To give intuition into the perturbation algorithm, let N be the number of users in the community. Let M be the number of data points shared by each user (we assume this to be the same across users for notational simplicity, but we shall develop algorithms that do not depend on that). Let $x^i = (x_1^i, x_2^i, \ldots, x_M^i)$, $n^i = (n_1^i, n_2^i, \ldots, n_M^i)$, and

$y^i = (y_1^i, y_2^i, \ldots, y_M^i)$ represent the data stream, noise, and perturbed data shared by user i, respectively. At time instant k, let $f_k(x)$ be the empirical community distribution, $f_k^e(x)$ be the exact community distribution, $f_k(n)$ be the noise distribution, and $f_k(y)$ be the perturbed community distribution. User data streams can be generated according to either linear or non-linear discrete models. In general, a model can be written as a discrete function of index k, which can be time, distance, or other (depending on the application), parameters $\underline{\boldsymbol{\theta}}$, and inputs $\underline{\mathbf{u}}$, and is denoted as $g(k, \underline{\boldsymbol{\theta}}, \underline{\mathbf{u}})$. Notice that $\underline{\boldsymbol{\theta}}$ is a fixed length parameters vector characterizing the model while $\underline{\mathbf{u}}$ is a vector of length M characterizing the input to the model at each instance. Given the data $x^i = (x_1^i, x_2^i, \ldots, x_M^i)$, the model $g(k, \underline{\boldsymbol{\theta}}, \underline{\mathbf{u}})$, and the approximated distributions $f_\theta^n(\underline{\boldsymbol{\theta}})$, $f_u^n(\underline{\mathbf{u}})$, the perturbed data for user i is generated by (i) generating samples $\underline{\boldsymbol{\theta}}_{\mathbf{n}}^{\mathbf{i}}$ and $\underline{\mathbf{u}}_{\mathbf{n}}^{\mathbf{i}}$, from the distributions $f_\theta^n(\boldsymbol{\theta})$ and $f_u^n(\underline{\mathbf{u}})$, respectively, (ii) generating noise stream $n^i = (n_1^i, n_2^i, \ldots, n_M^i)$, where $n_k^i = g(k, \underline{\boldsymbol{\theta}}_{\mathbf{n}}^{\mathbf{i}}, \underline{\mathbf{u}}_{\mathbf{n}}^{\mathbf{i}})$, and (iii) generating perturbed data by adding the noise stream to the data stream $y^i = x^i + n^i$.

Now, consider reconstructing the *distribution* of community data at a given point in time. At time instance k, the perturbed data of each user is the sum of the actual data and the noise $y_k^i = x_k^i + n_k^i$. Thus, the distribution of the perturbed data $f_k(y)$ is the *convolution* of the community distribution $f_k(x)$ and the noise distribution $f_k(n)$, $f_k(y) = f_k(n) * f_k(x)$.

All the distributions above can be discretized as:

$$
\begin{aligned}
f_k(n) &= (fn(0), fn(1), \ldots, fn(L)) \\
f_k(x) &= (fx(0), fx(1), \ldots, fx(L)) \\
f_k(y) &= (fy(0), fy(1), \ldots, fy(2L))
\end{aligned}
$$

The convolution can therefore be rewritten as:

$$
fy(m) = \sum_{k=-\infty}^{\infty} fx(k) fn(m-k) \tag{11}
$$

Since convolution is a linear operator, Eq. (11) can be written as

$$
f_k(y) = H f_k(x) \tag{12}
$$

where H is a $L \times (2L+1)$ Toeplitz cyclic matrix, which is also called the blurring kernel, constructed from elements of the discrete distribution $f_k(n)$ as:

$$
H = \begin{pmatrix}
fn(0) & 0 & 0 & \ldots & 0 \\
fn(1) & fn(0) & 0 & \ldots & 0 \\
fn(2) & fn(1) & fn(0) & \ldots & 0 \\
\ldots & \ldots & \ldots & \ldots & \ldots \\
0 & 0 & \ldots & fn(L) & fn(L-1) \\
0 & 0 & 0 & \ldots & fn(L)
\end{pmatrix} \tag{13}
$$

In Eq. (12), $f_k(x)$ is the community distribution at time k that needs to be estimated, H is known and $f_k(y)$ is the empirical perturbed data distribution. This problem is well known in the literature as the *deconvolution problem*.

Several algorithms have been developed to solve this problem and can be categorized into two classes. The first is a set of *iterative algorithms*, such as Richardson-Lucy algorithm, the EM algorithm, and the Poisson MAP method. The second class of algorithms are *non-iterative*. Examples include Tikhonov-Miller restoration and SECB restoration.

Consider the Tikhonov-Miller restoration [77], for an example. It requires an apriori bound ε for the L^2 norm of the noise, together with an apriori bound M for the L^2 norm of the community distribution:

$$||Hf_k^e(x) - f_k(y)||_2 \leq \varepsilon \tag{14}$$

$$||(H^T H)^{-\nu} f_k^e(x)||_2 \leq M \tag{15}$$

where $||\ \ ||_p$ denotes the $L_p(R)$ norm of a vector. The optimal solution $f_k(x)$ is chosen to minimize the regularized quadratic functional:

$$||Hf_k(x) - f_k(y)||_2^2 + \left(\frac{\varepsilon}{M}\right)^2 ||(H^T H)^{-\nu} f_k(x)||_2^2 \tag{16}$$

The fraction $\lambda = \varepsilon/M$ is called the *regularization coefficient* which governs the relative importance between the error and the regularized term.

By minimizing Eq. (16), the exact expression for the optimal solution $f_k^*(x)$ can be found:

$$f_k^*(x) = Q_T^{-1} H^T f_k(y) \tag{17}$$

$$Q_T = H^T H + \left(\frac{\varepsilon}{M}\right)^2 (H^T H)^{-2\nu} \tag{18}$$

Equations (17) and (18) can be used in an aggregation server to reconstruct the community distribution.

5.2 Evaluation

In recent work [66], a participatory sensing service was described where drivers are allowed to "lie" about their speed, location, and time, yet the service is able to construct accurate congestion maps that plot the right average traffic speed as a function of true location (street) and true time of day. This problem is related to the more general concern of privacy-preserving classification [85], except that it is applied to the challenging case of aggregates of time-series data. Figure 2 presents preliminary results comparing the real and estimated community speed distributions for different streets. Informally, it can be seen that these distributions are very close.

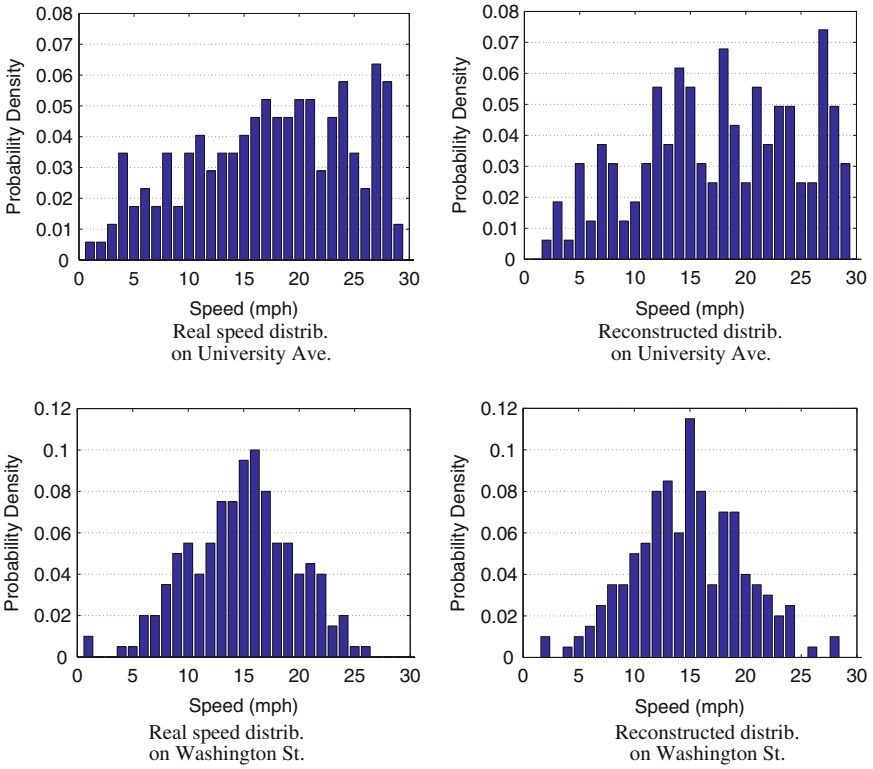


Fig. 2 Comparison of real and reconstructed speed distrib

Comparing the percentage of speeding vehicles computed from the real and perturbed distributions of different streets, we find that the error is small (e.g., University Ave 15.60 % vs 17.89 %, Neil Street 21.43 % vs 23.67 %, Washington Street 0.5 % vs 0.15 %, and Elm Street 6.95 % vs 8.6 %).

6 Challenge #3: Generalized Model Construction

Another social sensing challenge is model extraction from high-volume, sparse, high-dimensional raw data. Model extraction is difficult for two reasons. On one hand, the sheer volume of collected data at a server may quickly become overwhelming, hence requiring scalable solutions. On the other hand, data are usually parametrized by many attributes (e.g., measured fuel consumption of a vehicle depends on its make, model, class, year, and other vehicle parameters). This leads to a high-dimensional space that is sparsely populated due to the exponential explosion of attribute combinations. Deriving models for that space is challenging. While a large number of

efficient regression analysis and modeling techniques exist in current literature, a significant research problem is automated *partitioning* of the input attribute space such that one may derive accurate models with an appropriately small number of parameters in each subspace. This is akin to linearization of a complex system by a collection of linear models. One difference is that linearization only simplifies the modeling function, but does not change its inputs. Instead, our goal is to determine the best inputs as well, which may be different from one subspace to another. For example, when modeling human behavior, the best behavior predictors (inputs of the behavior model) might be substantially different depending on age. Hence, in high-dimensional data spaces, it is key to automatically jointly determine a good partitioning of the space and good models in each partition.

A recent solution approach leverages and extends OLAP (On-line analytical processing), a well-investigated topic in the database community [16, 36] when dealing with multi-dimensional data. The underlying data structure in OLAP systems is often called an *OLAP Cube* [36]. This cube can be thought of as a lattice where the root represents aggregate properties of the entire data set, and each level of descendants drills down by grouping the data by one additional dimension. This creates an exponential number of subcubes each grouping the data by a different subset of the original attributes. For example, the (car class, car year) subcube groups all data by car class and year. A specific instance such as (car class = midsize, car year = 2004) is called a cell. Operations common on data cells in commercial databases include sum and average [37]. Unfortunately, queries common to commercial database applications are different from the type of information one expects to learn from sensing the *physical world*. Physical data typically obey underlying physical models that we wish to extract. Hence, data in a cube should be organized in a way that facilitates model construction and analysis; rather than returning the average of a cell, one might want to return a physical model that fits cell data.

A new category of OLAP cubes is called the *sampled regression* and *sampled prediction* cubes. They extend the concepts of regression and prediction cubes covered in previous literature [17–19, 69], by considering issues of sparse deployment and hence sparse sampling inherent to socially-embedded cyber-physical systems. Current regression or prediction cubes compute, for each subcube, a regression or prediction model that fits cell data.[1] Partial deployment of new services gives rise to lengthy periods where many of the cells have no data or have a small amount of data that is not statistically significant [54]. Answering queries about such missing data requires generalization. The challenge of generalization from sparse data has been previously proposed in the context of sampled cubes, but used for the purpose of generalizing only simple cell statistics such as sum and average [54]. Generalizing regression and prediction models, such as prediction of fuel efficiency, is substantially more complicated and requires further advances in data cubes. A particular challenge

[1] Regression is a technique from estimation theory, applied to continuous or inherently ordered parameters to predict continuous or ordered values. In contrast, prediction uses machine learning to predict unordered class labels.

is to explore *automated* techniques for model construction that exloit correlations between parameters and compute their predictiveness to automate identification of the best model structure from the data cube. Below, we review one technique that achieves the above.

6.1 Computing Regression Models and Error

Consider the problem of optimally partitioning a data space into subspaces given by different regression models. We start by building a least squares estimate, in each cell c, that estimates a parameter vector, $\hat{\eta}_c$, and error Err_c, given a set of n_c data tuples, each with k inputs and one output, organized into arrays X_c and Y_c, respectively. This estimate is our regression model for the subspace, c. Note that a cell is the "unit of resolution" in subdividing the input space. One must then try different ways of coalescing cells that have similar models in hopes of reducing the total number of models used for different parts of the space, while simultaneously increasing statistical confidence in each. A main challenge is to express the standard regression model and its estimation error in a way that can be hierarchically computed, such that models and errors computed from larger data sets can be composed from those of their subsets without refering back to the original data. This recursion would significantly speed-up search for the best coalescing of similar cells that results in the most accurate and general models, or equivalently the best partitioning of the overall data space. Hence, an efficient "divide" process would be achieved.

Let us define a scalar $\rho_c = Y_c^T Y_c$, a vector (of size k) $\nu_c = X_c^T Y_c$, and a matrix (of size $k \times k$) $\Theta_c = X_c^T X_c$. Starting with the common expression for a least squares estimator, we can now reformulate the derivation of regression coefficients and error for cell c as follows:

$$\hat{\eta}_c = (X_c^T X_c)^{-1} X_c^T Y_c = \Theta_c^{-1} \nu_c \tag{19}$$

$$\begin{aligned} Err_c &= (Y_c - X_c\hat{\eta}_c)^T (Y_c - X_c\hat{\eta}_c) = Y_c^T Y_c - (X_c\hat{\eta}_c)^T Y_c - Y_c^T X_c\hat{\eta}_c + (X_c\hat{\eta}_c)^T X_c\hat{\eta}_c \\ &= \rho_c - \hat{\eta}_c^T \nu_c - \nu_c^T \hat{\eta}_c + \hat{\eta}_c^T \Theta_c \hat{\eta}_c \end{aligned} \tag{20}$$

What is significant about the above expressions is that our estimated model parameters, $\hat{\eta}_c$, and error, Err_c, are now expressed exclusively in terms of the intermediate quantities ρ_c, ν_c, and Θ_c. Note also that the dimensions of these intermediate quantities depend only on the number of regressors, independently of the input data size.

What is more significant, however, is that models of larger spaces can now be computed recursively from those of smaller subspaces. Let $i = 1, \ldots, m$ be the m cells used to obtain aggregate values for a cell c. It is easy to see that ρ_c, ν_c, and n_c are *distributive measures* and can be accurately aggregated as follows:

$$\rho_c = Y_c^T Y_c = [Y_{c_1} \dots Y_{c_m}] \left[Y_{c_1} \dots Y_{c_m}\right] = \sum_{i=1}^{m} Y_{c_i}^T Y_{c_i} = \sum_{i=1}^{m} \rho_{c_i}$$

$$\nu_c = X_c^T Y_c = [X_{c_1} \dots X_{c_m}] \left[Y_{c_1} \dots Y_{c_m}\right] = \sum_{i=1}^{m} X_{c_i}^T Y_{c_i} = \sum_{i=1}^{m} \nu_{c_i}$$

$$n_c = \sum_{i=1}^{m} n_{c_i} \qquad \Theta_c = \sum_{i=1}^{m} \Theta_{c_i}$$

Hence, having computed the above parameters (the algebraic measures) for each cell, one can determine the regression model and error for any aggregation of cells simply by adding the corresponding per-cell parameters and using Eqs. (19) and (20), respectively. The distributive property of the algebraic measures mentioned above allows efficient search for best generalizations.

6.2 Reliability Measure

A second challenge in modeling using regression cubes is to develop a *reliability criterion* which only uses the information stored in the data cells (i.e., the algebraic representation) in order to determine model reliability. Not requiring further information makes this criterion easy to evaluate and therefore usable in the context of a data cube with a potentially large number of data cells.

For the estimation of cell reliably as a function of algebraic measures, the reader is referred to recent work [9]. Below, we simply state the result. Namely, the prediction error in the cell remains below δ with probability $1-\varepsilon$, if $n_c > k$ and $\frac{k\sigma^2}{\delta^2 \lambda_{min}(\Theta_c)} < \varepsilon$, where λ_{min} denotes the minimum eigenvalue and $\sigma^2 = \frac{Err_c}{n_c - k}$. This equation allows testing whether the model computed in a cell is reliable in the sense of being bounded to a specified error δ with a specified probability $1-\varepsilon$. The contribution of the above expression lies in the fact that it is given exclusively in terms of distributive measures. Hence, cell reliability can be efficiently computed.

We can now summarize the cube construction procedure as follows: We start by the apex cuboid (which groups all data together and builds a single model) and compute the regression model for all of the cells in that cuboid and its children cuboids. We then find the cuboid with the lowest prediction error compared to its parent. Cuboids that contain an unreliable cell are discarded. At each step, we add the cuboid with the maximum reduction in average error and consider its children as the next-step candidates. This is done until there is no error reduction above some threshold, or all new cells are unreliable (indicating that we do not have sufficient data there). The result is a partitioning of the space that satisfies model reliability requirements and maximizes accuracy.

6.3 Evaluation

To apply the above concepts consider a green navigation application, called GreenGPS, in which it is desired to estimate the fuel consumption of a vehicle on any given path from a source to a destination. There are several factors that affect the fuel consumption of cars on streets. They can be classified into (i) *street parameters*, (ii) *car parameters*, and (iii) *personal parameters*. In GreenGPS, vehicle weight, size, tire diameter, engine type, gear ratio, shape, and manufacturer all play a role in fuel consumption. Given a high-dimensional space, we shall develop techniques that group cars into categories and find minimal-parameter models for each category, from collected data, that have a sufficiently accurate predictive power. Further, street parameters can refer to static street characteristics, such as speed limit and number of lanes, as well as dynamic parameters such as the actual average speed (recently made available on large city streets by a Google service that uses feeds from participating GPS devices in cell phones).

Figure 3 demonstrates the efficacy of sparse regression cubes at modeling the fuel consumption of a car, comparing them to three other approaches; namely (i) a single multi-dimensional regression model obtained by using support vector regression (SVR), (ii) regression cubes [18], and (iii) sampling cubes [54]. Figure 3a shows that the proposed technique significantly improves prediction accuracy. Figure 3b, shows that the analytically computed estimated error is very close to the actual observed error and that it is indeed less than the (95 %) confidence bound.[2] These results are encouraging, motivating further investigation into reliable modeling of physical systems from social sensing data.

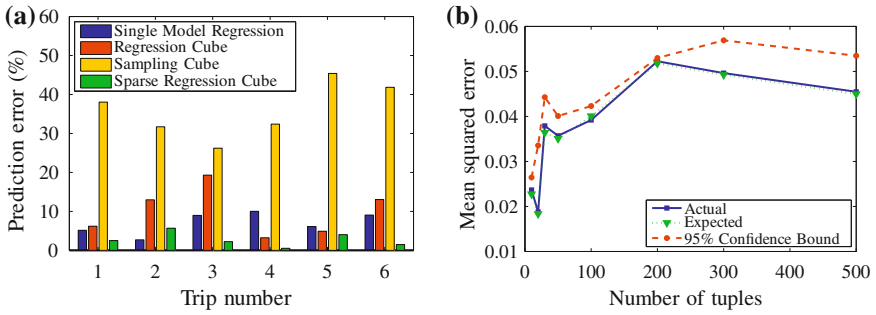


Fig. 3 Performance Evaluation **a** Prediction accuracy for 6 different trips. **b** The prediction confidence measured

[2] To make error values meaningful, we have normalized fuel consumption values to be zero mean and between −1 and 1.

7 Summary and Discussion

In this chapter, we reviewed a set of analytic challenges motivated by emerging social sensing systems. These include quantifying the correctness of collected data, ensuring data privacy, and developing appropriate models from data. Solutions to address these key challenge are non-trivial.

Social sensing systems are one example of *information distillation* systems. Information distillation (or reduction of large amounts of data into smaller amounts of actionable information) is an increasingly important interaction modality between humans and data. While the ability of technology to generate and disseminate information has grown dramatically in recent years, human cognitive ability to consume it remains unchanged. This imbalance between the growing availability of information and the human capacity to consume it suggests the increasing importance of the category of services whose main objective is to distill real-time information for human consumption.

Information distillation services are further made popular by a shift in the digital information landscape. This shift is from a web of slowly updated cross-linked objects (e.g., Web pages) to streams of continually generated real-time data emitted by humans and sensors. Examples include data shared on social networks such as Twitter, Flikr, YouTube, and FourSquare, and data generated by sensors such as neighborhood watch cameras and medical devices for longitudinal monitoring. The availability of real-time data offers both new opportunities and new challenges. There are unprecedented opportunities for building real-time "situation awareness" applications such as disaster-response services that help first-responders assess current damage, transportation advisories that help individuals avoid traffic bottlenecks, and citizen-science tools that collect and process data from speciality sensors (such as rain guages or pollution sensors) owned by interested individuals. Problems discussed in this chapter are the challenges that arise and are fundamental to information distillation. Indeed, reliable information must be distilled from unreliable data. Individuals who share the data may require some privacy guarantees. Applications that use the data must address modeling challenges, in order to offer predictive services that learn from past observations. Several categories of sensing applications were discussed ranging from those where humans collect data points that are individually significant to those where models or statistical properties of aggregates are sought.

Many other challenges remain topics of current research. Those include front-end challenges (e.g., energy consumption), coordination challenges (e.g., participatory sensing campaign recruitment), back-end challenges (e.g., modeling and prediction), and challenges in the overall understanding of the emergent behavior of social sensing systems at large. While a significant amount of research has already been undertaken along those fronts, much remains unsolved. New interdisciplinary research is needed to bring about better solutions for a better theoretical understanding of emerging social sensing systems in a future sensor- and media-rich world.

References

1. T. Abdelzaher et al., Mobiscopes for human spaces. IEEE Pervasive Comput. **6**(2), 20–29 (2007)
2. T.F. Abdelzaher, Y. Anokwa, P. Boda, J. Burke, D. Estrin, L.J. Guibas, A. Kansal, S. Madden, J. Reich, Mobiscopes for human spaces. IEEE Pervasive Comput. **6**(2), 20–29 (2007)
3. K. Aberer, Z. Despotovic, Managing trust in a peer-2-peer information system. in *CIKM '01: Proceedings of the Tenth International Conference on Information and Knowledge Management*, pp. 310–317. ACM, New York, NY, USA, 2001
4. G. Adomavicius, A. Tuzhilin, Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. IEEE Trans. Knowl. Data Eng. **17**(6), 734–749 (2005)
5. C. Aggarwal, T. Abdelzaher, Social sensing. in *Managing and Mining Sensor Data* (Kluwer Academic Publishers, Boston, 2013)
6. C. Aggarwal, T. Abdelzaher, Integrating sensors and social networks. *Social Network Data Analytics* (Springer, expected in 2011)
7. D. Agrawal, C.C. Aggarwal, On the design and quantification of privacy preserving data mining algorithms. in *Proceedings of the 20th ACM SIGMOD Symposium on Principles of Database Systems*, pp. 247–255, 2001
8. R. Agrawal, R. Srikant, Privacy preserving data mining. in *Proceedings of ACM Conference on Management of Data*, pp. 439–450, May 2000
9. H. Ahmadi, T. Abdelzaher, J. Han, N. Pham, R.K. Ganti. The sparse regression cube: a reliable modeling technique for open cyber-physical systems. in *Proceedings of the 2011 IEEE/ACM Second International Conference on Cyber-Physical Systems, ICCPS '11*, pp. 87–96. IEEE Computer Society, Washington, DC, USA, 2011
10. H. Ahmadi, N. Pham, R. Ganti, T. Abdelzaher, S. Nath, J. Han, Privacy-aware regression modeling of participatory sensing data. in *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems, SenSys '10*, pp. 99–112. ACM, New York, NY, USA, 2010
11. R. Balakrishnan, Source rank: relevance and trust assessment for deep web sources based on inter-source agreement. in *20th World Wide Web Conference (WWW'11)* 2011
12. L. Berti-Equille, A.D. Sarma, X. Dong, A. Marian, D. Srivastava, Sailing the information ocean with awareness of currents: discovery and application of source dependence. in *CIDR'09* 2009
13. J. Bilmes, A gentle tutorial on the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. Technical report, University of Berkeley, ICSI-TR-97-021, 1997
14. A.T. Campbell, S.B. Eisenman, N.D. Lane, E. Miluzzo, R.A. Peterson, H. Lu, X. Zheng, M. Musolesi, K. Fodor, G.-S. Ahn, The rise of people-centric sensing. IEEE Internet Comput. **12**(4), 12–21 (2008)
15. G. Casella, R. Berger, *Statistical Inference* (Duxbury Press, Pacific Grove, 2002)
16. S. Chaudhuri, U. Dayal, An overview of data warehousing and OLAP technology. SIGMOD Rec. **26**, 65–74 (1997)
17. B.-C. Chen, L. Chen, Y. Lin, R. Ramakrishnan, Prediction cubes. in *Proceedings 2005 International Conference Very Large Data Bases (VLDB'05)*, pp. 982–993, Trondheim, Norway, Aug 2005
18. Y. Chen, G. Dong, J. Han, J. Pei, B.W. Wah, J. Wang, Regression cubes with lossless compression and aggregation. IEEE Trans. Knowl. Data Eng. **18**, 1585–1599 (2006)
19. Y. Chen, G. Dong, J. Han, B. W. Wah, J. Wang. Multi-dimensional regression analysis of time-series data streams. in *Proceedings 2002 International Conference Very Large Data Bases (VLDB'02)*, pp. 323–334, Hong Kong, China, Aug 2002
20. D.J. Cook, L.B. Holder, Sensor selection to support practical use of health-monitoring smart environments. Wiley Interd. Rev. Data Min. Knowl. Discovery **1**(4), 339–351 (2011)
21. H. Cramer. *Mathematical Methods of Statistics* (Princeton University Press, Princeton, 1946)

22. O. Dekel, O. Shamir, Vox populi: collecting high-quality labels from a crowd. in *In Proceedings of the 22nd Annual Conference on Learning Theory* 2009
23. S.A. Delre, W. Jager, M.A. Janssen, Diffusion dynamics in small-world networks with heterogeneous consumers. Comput. Math. Organ. Theory **13**, 185–202 (2007)
24. A.P. Dempster, N.M. Laird, D.B. Rubin, Maximum likelihood from incomplete data via the em algorithm. J. Roy. Stat. Soc. B **39**(1), 1–38 (1977)
25. X. Dong, L. Berti-Equille, Y. Hu, D. Srivastava, Global detection of complex copying relationships between sources. PVLDB **3**(1), 1358–1369 (2010)
26. X. Dong, L. Berti-Equille, D. Srivastava, Truth discovery and copying detection in a dynamic world. VLDB **2**(1), 562–573 (2009)
27. A. Doucet, N. De Freitas, N. Gordon, (eds.), *Sequential Monte Carlo Methods, in Practice* (Springer, New York, 2001)
28. R.O. Duda, P.E. Hart, D.G. Stork. *Pattern Classification*, 2nd edn. (Wiley-Interscience, New York, 2001)
29. S.B. Eisenman et al., The bikenet mobile sensing system for cyclist experience mapping. in *Proceedings of SenSys*, Nov 2007
30. A. Evfimievski, J. Gehrke, R. Srikant, Limiting privacy breaches in privacy preserving data mining. in *Proceedings of the SIGMOD/PODS Conference*, pp. 211–222, 2003
31. A. Galland, S. Abiteboul, A. Marian, P. Senellart, Corroborating information from disagreeing views. in *WSDM*, pp. 131–140, 2010
32. R. Ganti, N. Pham, Y.-E. Tsai, T. Abdelzaher, Poolview: stream privacy for grassroots participatory sensing. in *ACM Sensys*, Raleigh, NC, Nov 2008
33. R.K. Ganti, S. Srinivasan, A. Gacic, Multisensor fusion in smartphones for lifestyle monitoring. in *Proceedings of the 2010 International Conference on Body Sensor Networks, BSN '10*, pp. 36–43. IEEE Computer Society, Washington, DC, USA, 2010
34. P. Gilbert, L.P. Cox, J. Jung, D. Wetherall, Toward trustworthy mobile sensing. in *Proceedings of the Eleventh Workshop on Mobile Computing Systems and Applications, HotMobile '10*, pp. 31–36. ACM, New York, NY, USA, 2010
35. P. Gilbert, J. Jung, K. Lee, H. Qin, D. Sharkey, A. Sheth, L.P. Cox. Youprove: authenticity and fidelity in mobile sensing. in *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems, SenSys '11*, pp. 176–189. ACM, New York, NY, USA, 2011
36. J. Gray, S. Chaudhuri, A. Bosworth, A. Layman, D. Reichart, M. Venkatrao, F. Pellow, H. Pirahesh, Data cube: a relational aggregation operator generalizing group-by, cross-tab and sub-totals. Data Min. and Knowl. Discovery **1**, 29–54 (1997)
37. V. Harinarayan, A. Rajaraman, J.D. Ullman. Implementing data cubes efficiently. in *Proceedings 1996 ACM-SIGMOD International Conference Management of Data (SIGMOD'96)*, pp. 205–216, Montreal, Canada, June 1996
38. A. Helal, D.J. Cook, M. Schmalz, Smart home-based health platform for behavioral monitoring and alteration of diabetes patients. J. Diab. Sci. Technol. **3**(1), 141–148 (2009)
39. K. Hoffman, D. Zage, C.N. Rotaru, A survey of attack and defense techniques for reputation systems. ACM Comput. Surv. **42**(1), 1–31 (2009)
40. R.V. Hogg, A.T. Craig, *Introduction to Mathematical Statistics* (Prentice Hall, Upper Saddle River, 1995)
41. D. Houser, J. Wooders, Reputation in auctions: theory, and evidence from ebay. J. Econ. Manage. Strategy **15**(2), 353–369 (2006)
42. J. Huang. Color-spatial image indexing and applications. Ph.D. thesis, Cornell University, 1998
43. J.-H. Huang, S. Amjad, S. Mishra, Cenwits: a sensor-based loosely coupled search and rescue system using witnesses. in *Proceedings of SenSys*, pp. 180–191, 2005
44. Z. Huang, W. Du, B. Chen, Deriving private information from randomized data. in *Proceedings of the 2005 ACM SIGMOD Conference*, pp. 37–48, Baltimore, MD, June 2005
45. C. Hui, M.K. Goldberg, M. Magdon-Ismail, W.A. Wallace, Simulating the diffusion of information: an agent-based modeling approach. in *IJATS*, pp. 31–46, 2010
46. B. Hull et al. Cartel: a distributed mobile sensor computing system. in *Proceedings of SenSys*, pp. 125–138, 2006

47. U.T. Inc, U.T.I. Staff, *Solving Data Mining Problems Using Pattern Recognition Software with CDROM*, 1st edn. (Prentice Hall PTR, Upper Saddle River, 1997)
48. J. Han, M. Kamber, J. Pei, *Data Mining: Concepts and Techniques*, 3rd edn. (Morgan Kaufman, San Francisco, 2011)
49. R.A. Johnson, D.W. Wichern, *Applied Multivariate Statistical Analysis* (Prentice-Hall, Inc., Upper Saddle River, 2002)
50. A. Jøsang, R. Ismail, C. Boyd, A survey of trust and reputation systems for online service provision. Decis. Support Syst. **43**(2), 618–644 (2007)
51. R.E. Kalman, A new approach to linear filtering and prediction problems. Trans. ASME J. Basic Eng. **82**(Series D), 35–45 (1960)
52. H. Kargutpa, S. Datta, Q. Wang, K. Sivakumar, On the privacy preserving properties of random data perturbation techniques. in *Proceedings of the IEEE International Conference on Data Mining*, pp. 99–106, 2003
53. J.M. Kleinberg, Authoritative sources in a hyperlinked environment. J. ACM **46**(5), 604–632 (1999)
54. X. Li, J. Han, Z. Yin, J.-G. Lee, Y. Sun, Sampling cube: a framework for statistical OLAP over sampling data. in *Proceedings 2008 ACM SIGMOD International Conference on Management of Data (SIGMOD'08)*, Vancouver, BC, Canada, June 2008
55. Q. Lian, Z. Zhang, M. Yang, B. Y. Zhao, Y. Dai, X. Li, An empirical study of collusion behavior in the maze p2p file-sharing system. in *Proceedings of the 27th International Conference on Distributed Computing Systems, ICDCS '07*, p. 56. IEEE Computer Society, Washington, DC, USA, 2007
56. B. Longstaff, S. Reddy, D. Estrin, Improving activity classification for health applications on mobile devices using active and semi-supervised, learning, p. 6, 2010
57. A. Madan, M. Cebrian, D. Lazer, A. Pentland, Social sensing for epidemiological behavior change. in *Proceedings of the 12th ACM International Conference on Ubiquitous Computing, Ubicomp '10*, pp. 291–300. ACM, New York, NY, USA, 2010
58. A. Madan, S.T. Moturu, D. Lazer, A. Pentland, Social sensing: obesity, unhealthy eating and exercise in face-to-face networks. in *Wireless, Health*, pp. 104–110, 2010
59. G.J. McLachlan, T. Krishnan, *The Em Algorithm and Extensions* (Wiley, New York, 1997)
60. E. Miluzzo, N.D. Lane, K. Fodor, R. Peterson, H. Lu, M. Musolesi, S.B. Eisenman, X. Zheng, A.T. Campbell, Sensing meets mobile social networks: the design, implementation and evaluation of the cenceme application. in *Proceedings of the 6th ACM conference on Embedded network sensor systems, SenSys '08*, pp. 337–350. ACM, New York, NY, USA, 2008
61. M. Mun, S. Reddy, K. Shilton, N. Yau, J. Burke, D. Estrin, M. Hansen, E. Howard, R. West, P. Boda, Peir, the personal environmental impact report, as a platform for participatory sensing systems research. in *Proceedings of the 7th International Conference on Mobile Systems, Applications, and Services, MobiSys '09*, pp. 55–68. ACM, New York, NY, USA, 2009
62. N. Mustapha, M. Jalali, M. Jalali, Expectation maximization clustering algorithm for user modeling in web usage mining systems. Eur. J. Sci. Res. **32**(4), 467–476 (2009)
63. S. Nath, Ace: exploiting correlation for energy-efficient and continuous context sensing. in *Proceedings of the Tenth International Conference on Mobile systems, Applications, and Services (MobiSys'12)* 2012
64. T. Park, J. Lee, I. Hwang, C. Yoo, L. Nachman, J. Song, E-gesture: a collaborative architecture for energy-efficient gesture recognition with hand-worn sensor and mobile devices. in *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems, SenSys '11*, pp. 260–273. ACM, New York, NY, USA, 2011
65. J. Pasternack, D. Roth, Knowing what to believe (when you already know something). in *International Conference on Computational Linguistics (COLING)* 2010
66. N. Pham, R. Ganti, M.Y. Uddin, S. Nath, T. Abdelzaher, Privacy-preserving reconstruction of multidimensional data maps in vehicular participatory sensing. in *EWSN*, Coimbra, Portugal, Feb 2010
67. N. Pham, R.K. Ganti, Y.S. Uddin, S. Nath, T. Abdelzaher, Privacy-preserving reconstruction of multidimensional data maps in vehicular participatory sensing. in *Proceedings of the 7th*

European Conference on Wireless Sensor Networks, EWSN'10, pp. 114–130. Springer-Verlag, Berlin, Heidelberg, 2010
68. D. Pomerantz, G. Dudek, Context dependent movie recommendations using a hierarchical bayesian model. in *Proceedings of the 22nd Canadian Conference on Artificial Intelligence: Advances in Artificial Intelligence, Canadian AI '09*, pp. 98–109. Springer-Verlag, Berlin, Heidelberg, 2009
69. R. Ramakrishnan, B.-C. Chen, Exploratory mining in cube space. Data Min. Knowl. Discovery **15**, 29–54 (2007)
70. S. Reddy, D. Estrin, M. Srivastava, Recruitment framework for participatory sensing data collections. in *Proceedings of the 8th International Conference on Pervasive Computing*, pp. 138–155. Springer, Berlin, Heidelberg, May 2010
71. S. Reddy, K. Shilton, G. Denisov, C. Cenizal, D. Estrin, M. Srivastava, Biketastic: sensing and mapping for better biking. in *Proceedings of the 28th International Conference on Human Factors in Computing Systems, CHI '10*, pp. 1817–1820. ACM, New York, NY, USA, 2010
72. Sense Networks. Cab Sense. http://www.cabsense.com
73. V.S. Sheng, F. Provost, P.G. Ipeirotis, Get another label? Improving data quality and data mining using multiple, noisy labelers. in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '08*, pp. 614–622. ACM, New York, NY, USA, 2008
74. Y. Sun, Y. Yu, J. Han, Ranking-based clustering of heterogeneous information networks with star network schema. in *15th SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'09)*, pp. 797–806, 2009
75. P.-N. Tan, M. Steinbach, V. Kumar, *Introduction to Data Mining* (Addison Wesley, Boston, 2005)
76. A. Thiagarajan, J. Biagioni, T. Gerlich, J. Eriksson, Cooperative transit tracking using smartphones. in *SenSys'10*, pp. 85–98, 2010
77. A.N. Tikhonov, V.Y. Arsenin, *Solution of Ill Posed Problems* (V. H. Winstons and Sons, Washington, 1977)
78. D. Wang, T. Abdelzaher, H. Ahmadi, J. Pasternack, D. Roth, M. Gupta, J. Han, O. Fatemieh, H. Le, On bayesian interpretation of fact-finding in information networks. in *14th International Conference on Information Fusion (Fusion 2011)*, 2011
79. D. Wang, H. Ahmadi, T. Abdelzaher, H. Chenji, R. Stoleru, C. Aggarwal, Optimizing quality-of-information in cost-sensitive sensor data fusion. in *IEEE 7th International Conference on Distributed Computing in Sensor Systems (DCoSS 11)*, June 2011
80. D. Wang, L. Kaplan, T. Abdelzaher, C. Aggarwal, On scalability and robustness limitations of real and asymptotic confidence bounds in social sensing. in *9th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, Seoul, Korea, June 2012
81. D. Wang, H. Le, L. Kaplan, T. Abdelzaher. On truth discovery in social sensing: a maximum likelihood estimation approach. in *11th ACM/IEEE Conference on Information Processing in Sensor Networks (IPSN)*, April 2012
82. M.E. Whitman, H.J. Mattord, *Principles of Information Security* (Course Technology Press, Boston, 2004)
83. C.F.J. Wu, On the convergence properties of the EM algorithm. Ann. Stat. **11**(1), 95–103 (1983)
84. J. Xie, S. Sreenivasan, G. Korniss, W. Zhang, C. Lim, B.K. Szymanski, Social consensus through the influence of committed minorities. *CoRR*, abs/1102.3931, 2011
85. Z. Yang, S. Zhong, R.N. Wright, Privacy-preserving classification of customer data without loss of accuracy. in *Proceedings of SIAM International Conference on Data Mining*, pp. 92–102, 2005
86. X. Yin, J. Han, P.S. Yu, Truth discovery with multiple conflicting information providers on the web. IEEE Trans. Knowl. Data Eng. **20**, 796–808 (2008)
87. X. Yin, W. Tan, Semi-supervised truth discovery. in *WWW*. ACM, New York, NY, USA, 2011

88. H. Yu, M. Kaminsky, P.B. Gibbons, A. Flaxman, Sybilguard: defending against sybil attacks via social networks. SIGCOMM Comput. Commun. Rev. **36**, 267–278 (2006)
89. C. Zhai, A note on the expectation maximization (em) algorithm. *University of Illinois at Urbana Champaign, Department of Computer Scinece*, 2007
90. B. Zhao, B.I.P. Rubinstein, J. Gemmell, J. Han, A Bayesian approach to discovering truth from conflicting sources for data integration. Proc. VLDB Endow. **5**(6), 550–561 (2012)

Chapter 17
Behavior-Aware Mobile Social Networking

Wei-Jen Hsu and Ahmed Helmy

Abstract The next frontier in sensor networks is sensing the human society. Human interaction, with technology and within mobile communities provides enormous opportunities to provide new paradigms of user communication. Traditionally, communication in computer networks has focused on delivering messages to machine identities. Each host is uniquely addressed, and network protocols aim to find routes to a given machine identity efficiently. While this framework has been proven successful in the past, it is questionable whether it will be sufficient in the era of social networking and mobility. As we envision the emergence of mobile terminals tightly coupled with their users and thus reflect the behavior and preferences of the users, it is beneficial to consider an alternative (and complementary) framework: Could user behavior be collected and summarized as a representation of the user's interest, and be leveraged as a way to guide message delivery? In this chapter, we elaborate on this possibility, discussing user behavior trace collection, representation, and pioneering works on behavior-aware mobile network protocols.

This proposed new framework is to be used mainly as an alternative of the IP (routing) layer in the Internet today and provides a new mechanism for network message routing. However, as opposed to the current routing schemes (e.g., IP) which address each host with a unique ID, in this new framework it is the behavior descriptors of the hosts, not its identities, to be used as the target for a message. Therefore, in behavior-aware routing protocols, messages are destined to a behavior descriptor and it is moved across the network based on comparisons of behavior descriptor of intermediate nodes to the target behavior. Note that a behavior descriptor can map to many potential recipients, or none.

W.-J. Hsu (✉)
University of Florida, Fremont, CA, USA
e-mail: wjhsu@ufl.edu

A. Helmy
University of Florida, Gainesville, FL, USA
e-mail: helmy@cise.ufl.edu

H. M. Ammari (ed.), *The Art of Wireless Sensor Networks*,
Signals and Communication Technology, DOI: 10.1007/978-3-642-40066-7_17,

This chapter provides a survey of important research work on behavior-aware routing. In this chapter, we motivate and introduce the new paradigm in Sect. 1. In Sect. 2, we introduce the goal of behavior-aware routing and its challenges. We then introduce a framework, namely TRACE, to discuss the steps involved in the design of social behavior-aware routing. We also give examples from the literature to explain what each step involves. The most important task in this paradigm is to summarize and represent node behavior in a succinct form, in such a way that the new representation can be used in place of node identities (e.g., addresses) for routing. We then provide examples for various behavior-aware routing protocols from the literature in Sect. 3. Important research topics in this area for further study are discussed in Sect. 4. Section 5 concludes the chapter.

1 Why Behavior-Aware Social Networking?

In recent years, the rapid advances in wireless communication (e.g., Wifi, 3GPP standards for cellular data connectivity, WiMax) have made ubiquitous network connectivity an emerging reality. We are now free from the wires and can get network connectivity almost everywhere; at home, at school, at public hubs, and even on the go. More importantly, the mass production and adoption of portable devices (e.g., laptop computers, tablets, smart phones) have shifted the paradigm of how people access information and manage personal identity or presence on the Internet. Until just recently, Internet users used to share devices (e.g., a desktop computer at home or in the library) and access the Internet only from a limited number of information hubs. This is no longer the status quo, and will be even less so as we embrace further advances in mobile computing and communication devices.

This shift of paradigm has many interesting consequences. One particular noteworthy change is now people are in possession of personal communication/computing devices (e.g., PDAs, smart phones, laptops) which are used almost exclusively by their sole owners. This one-to-one tie between devices and owners has never been so direct and inseparable in the past. Due to this tie, it has become increasingly obvious that the way each mobile computing device is used is strongly influenced by its owner, reflecting her behavior and preferences directly. We can easily imagine, while many devices may be equally capable of accessing any information on the Internet from any physical location, a laptop owned by a computer science graduate student who is mostly on campus is likely to be used in significantly different ways than a smart phone owned by a business manager who travels around the globe. The difference manifests itself in many possible instances, including the locations and means by which the devices access the network, the physical mobility pattern, the applications used, the websites and data downloaded, other devices with which a peer-to-peer communication link is established, and many more. The regularity and repetition of the above behavioral aspects (i.e., how "predictable" the user is) also differ between different types of users.

Furthermore, emerging mobile devices today (e.g., smart phones) are equipped with unprecedented capabilities in terms of communication, computation, storage, and sensing. These capabilities enable mobile devices to sense, store, process, and exchange behavioral profiles that reflect detailed history of user activities and interactions. This facilitates customized experience for the users based on their inferred behavioral profiles.

As such, the mobile devices carried by its users could be conceptually considered as a sensor network of a different form. While the devices are mainly used for tasks other than sensing, they do have the capability to capture many aspects of their owner's daily lives. Today, the ever-increasing number of mobile devices in service form an ever-expanding sensing infrastructure capturing the human behavior. With this new opportunity, it is very possible to incorporate the sensed human behavior into the design of more efficient mobile networks.

This linkage between devices and owners further opens the door for the study of usage logs from mobile devices to understand the behavioral patterns of their owners. The analysis can be useful in many ways from a system-wide optimization to personalized recommendations. For example, an analysis of user mobility patterns on a university campus (e.g., summarizing when and where users log on to campus WLAN) can help the network administrator to understand the composition of network users and plan for future network deployments [8]. Analysis of websites one particular device accessed in the past can be used to construct better personalized data caching and recommendation systems. This direction is generally known as the "Behavior-Aware Mobile Networks" and is a research area evolving rapidly.

In this chapter, we are particularly interested in the concept of social-behavior aware message routing in mobile networks. Its basic concept is actually not very different from how information is spread among human beings without modern technology. Take a brief moment to reflect on how we acquire or spread information among people around us—In our social networks, each friend and acquaintance plays a different role, based on the closeness of the mutual relationship, personal interests and preferences, and many other factors. The friend you turn to for tips on online games may be different from the mentor you consult with for career advice. You spread recent discoveries on algorithms or on a new place to hang out among different social circles, based on their respective interests. We all observe and leverage our personal contacts according to their properties to achieve efficient outcomes in terms of acquiring or propagating information. Now, thanks to the personalization of mobile computing devices, a similar concept can also be instantiated in mobile networks formed by device peer-to-peer wireless radio connections. Since now each device reflects the personality and interests of its owner, it is very possible that a communication protocol observes and leverages this fact, and therefore makes the best use of the right contacts when sending or retrieving information. The key of success in social behavior-aware routing is to leverage the diversity in user behavior and choose the "proper" ones based on the nature and context of messages.

As an overview, design of a behavior-aware routing protocol involves the follow conceptual steps: (1) Behavior collection: each node keeps track of its own behavior and stores relevant events locally. (2) Behavior summarization and extraction: based

on the context of the problem, each node extracts a useful and succinct behavior representation from the events collected. (3) Behavior exchange and comparison: nodes then exchange their behavior representation, based on which they determine which nodes are useful to relay a particular message. (4) Message delivery: message are then transferred from one node to another, according to the choice made by the protocol based on behavior representations. Note that the above steps have to be carried out locally on each participating node, without a centralized entity which controls the network (which is impractical due to the dynamic nature of mobile networks).

There are many potential benefits in social behavior-aware routing, including: (1) Improving the success rate of acquiring the information wanted: it is always the best to "ask the experts" for each topic, and who the experts are of course depends heavily on the topic. (2) Increasing the visibility of a broadcast message with lower overhead: in a social network, there are usually people who represent information hubs. It is the best to broadcast messages through them to maximize message spread. (3) Reaching the right audience: if messages are delivered selectively, we minimize random messages that do not fit into our interest or focus. One person's spam may be the other person's treasure. (4) Providing anonymity: since messages are delivered based on the behavior descriptor of the devices, not the actual device identity, it is possible through careful protocol design to maintain anonymity of the true identity of the users. Note that there are typically many people with similar behavior at the level behavior representation is made, it is not possible to distinguish these users, helping anonymity.

2 Objective and Challenge in Behavior-Aware Routing

In this chapter, our specific goal is to utilize user behavior representation for message routing. To achieve this goal, we need to carefully understand the relationship between the target behavior for the message and the behavior that enables opportunities of communication. In mobile networks, typically, mobility of nodes creates "encounters" when two nodes are in physical proximity and they can exchange messages during this period over short-range radio (e.g., WiFi, Bluetooth, etc.). Usually, a representation in the realm of user mobility (e.g., mobility preference vector, encounter frequency vector, etc.) is used for nodes to estimate or predict potential future encounters. In general, nodes with similar mobility preference are more likely to encounter with each other; and due to natural tendency of human beings to have repetitive behavioral patterns in life, frequent past encounters is a good indicator for likelihood of future encounters. (See [12] for research on location preferences and repetitive patterns of mobile users).

There are two different objectives in using user behavior for routing: (1) Use user behavior to identify good candidate(s) to deliver a message to a particular node ID, and (2) use a user behavior descriptor itself as a target profile, and the protocol is responsible for both discovering nodes following this behavior, and delivering the message to them.

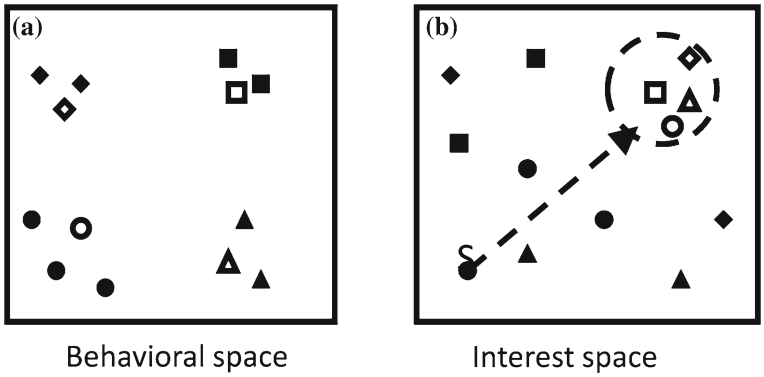


Fig. 1 **a** In behavioral space, nodes with similar mobility (nodes with the same legends) are close to each other. **b** In interest space, nodes with similar interest do not necessarily have similar mobility. (i.e., not of the same legend)

If the goal is to deliver a message to a particular node ID, the above construct of mobility preference or encounter probability is in general adequate for the sender or message relaying nodes to find out which contacts are more probable to deliver the message with higher success rate. This is illustrated in Fig. 1a, where the distance of the legends represents the similarity of their behavior. In the behavior space, nodes with similar mobility behavior (represented by the same legend) are more likely to encounter directly. Thus, similarity in behavior representations can be used as indicators of suitability to deliver a message, if the destination node's behavior representation is known. However, as was pointed out in the introduction, a major advantage of social behavior-aware routing is the potential paradigm shift to delivering the message based on user behavior. In this more generic setting, the goal of message delivery may or may not be related with the mobility-based representation. For example, a message sender may want to reach all classic music lovers in a community, but these potential recipients have various mobility preferences. In the more generic interest space, similar interest does not always lead to similar mobility patterns and thus higher chance to meet. It is therefore a more challenging problem to deliver messages based on mobility-independent interests, as illustrated in Fig. 1b. While some recent work in the literature attempts to start to bridge this gap, we believe social behavior aware routing in this more generic form still needs much further research.

In the following section we discuss a framework for steps necessary to achieve the goal listed in this section, including behavior collection, summarization, and validation.

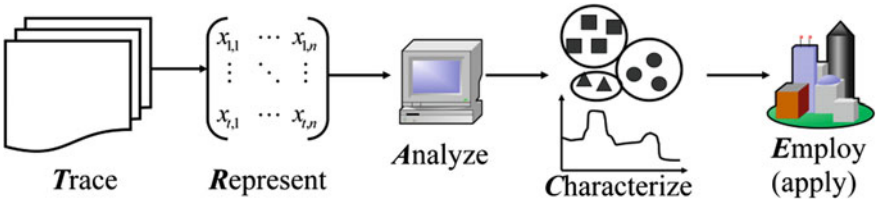


Fig. 2 Illustration of TRACE framework

3 Framework for Social Behavior Extraction and Examples

The foundation of social-aware routing roots in the collection and abstraction of user behavior data. Based on realistic data, detailed user behaviors can be extracted and leveraged for routing tasks. To this effect, we use the *TRACE* framework in Fig. 2 to outline the procedural steps for a study based on user data [10]. The study begins with ***T***race collection, followed by a series of procedures to ***R***epresent the data to capture meaningful and important behavioral trends. Then, further effort is exerted to ***A***nalyze and ***C***haracterize the users based on the representation, in order to identify metrics and structures useful for the application at hand. Finally, all the above understanding and insights are ***E***mployed in the actual application. We will describe each step in more depth below.

3.1 Collecting and Sharing of Behavioral Data

The first step towards behavior-aware mobile networking is the extensive collection of user traces. To achieve this goal, we leverage the existing infrastructure (e.g., mobile devices, WLAN access points, cellular phone towers) today as a sensor network to collect user behavior traces. The direct linkage between the mobile devices (e.g., laptops, smart phones, PDAs) and their owners today provides an excellent opportunity for the researchers to sense and capture human behavior from these devices owners closely.

User behavior is a very generic term. In a broad sense, it entails anything users do while they are online. Theoretically, to capture all aspects of behavior, one would keep as detailed as possible log of all operations a user performs, following every step the user takes as he uses the mobile device. However, not all aspects of user behavior are directly related to routing, and making very detailed user information available to the system operator and even to other users certainly raises privacy concerns. It is thus important to strike a balance between publishing sufficient information to enable behavior-aware routing, and adequately hiding information to ensure user privacy.

To solve this dilemma, the usual approach taken is the following: While collecting user data, very detailed information is retained or anonymized. However, before

such data is shared and made available, a pre-processing step will occur. This pre-processing step summarizes the data in a way that reduces the amount of information shared. As we will see later in the chapter, this operation not only helps to preserve user privacy, but also makes the raw data even more useful in a refined, summarized form.

In terms of data collection, there are three broad approaches: (1) Centralized approach, (2) Completely decentralized approach, and (3) User-aided centralized (hybrid) approach. There are a multitude of wireless network user/device activity traces available in the research community. Many of them are enlisted in trace archives, such as the CRAWDAD [2] or MobiLib [18] projects. We discuss a handful of prominent traces along with the approaches used to collect them below.

As a final note, although behavior traces can be collected in various ways to facilitate behavior-aware routing research, the actual working assumption of a behavior-aware routing protocol is usually that nodes need to collect its own behavior in a decentralized fashion. There is therefore no one entity that collects and stores the behavior data of all users. Centralized trace collection mentioned below is exclusively for the purpose of understanding how behavior-aware routing would work based on how users behave today.

3.1.1 Centralized (Infrastructure-Based) Trace Collection

Centralized data collection approach is the most commonly used in research literature, since it is a simple scheme to deploy at scale. In this approach, the individual users do not participate in data collection actively; they are passively monitored for their behavior, in most times not even knowing that such monitoring has occurred. One very common technique used in the research literature is to collect WLAN traces from access points to observe the behavior of users associated with the wireless network (e.g., when and where users log on, log off, how much data is transmitted, etc.). This approach does not require any changes to the users' devices, and thus can be carried out to collect traces from the general public once the infrastructure is in place. This makes the centralized monitoring approach the easiest to implement and scale, and the least intrusive (i.e., the users do not deviate from their normal behavior since they are usually not conscious about being monitored, even though they agree to the terms and conditions of the WLAN when they log in).

As an example for centralized data collection, the most extensive publicly available wireless network user logs are from wireless LANs (WLAN). The most common form of this type of trace logs the user association and disassociation events with access points (APs), and therefore can be used as a coarse-grained location log of the mobile nodes. WLAN logs have been collected from Dartmouth College, University of Southern California, University of Florida, IBM Watson research, EPFL, to name a few. A typical example of WLAN association traces is shown in Table 1. Some of these traces have additional information available. For example, the Dartmouth and IBM traces have syslogs that record the amount of traffic each AP sent and received. The USC trace has netflow data that keeps the flows generated by each user.

Table 1 Typical example of WLAN association trace

Node ID (Anonymized)	Access point location	Association start time	Association duration
A	Building X	326674	634
A	Building Y	327897	3294
B	Building X	328623	6298
C	Building Z	328821	2631
D	Building Y	329125	3742

Note that additional information may be available depending on the trace collection details

However, the centralized approach also has a strong dependency on the technology available today (e.g., WLAN), and there would always be the question of whether behavior observed from the trace is really an artifact of the technology itself, and not the intrinsic user behavior had the technology not been deployed. One classic example of such a debate is, while we can use the AP association in WLAN traces to observe how mobile devices move on a university campus, would this mobility pattern be the same if there were no WLAN deployed on campus?

3.1.2 Decentralized (User-Based) Trace Collection

On the contrary, the decentralized scheme usually requires each device to log its own traces, and later reports the log to a data repository. It is a more typical approach for small-scale, experimental trace collection. Usually, a specialized device or software is deployed through a carefully chosen, small set of subjects, and the goal of trace collection is focused on very specific aspects of user behavior. While the decentralized approach is more suitable for a specific need that cannot be covered by existing, deployed technology, it is also usually less scalable and can only be carried out in a much smaller and well-defined controlled set of subjects. Two prominent examples of decentralized trace collection from the literature are:

- The Haggle project (for pocket switched networks) [7] has set the focus on collecting human encounter patterns. They distributed iMotes (a small sensor with short-range radio) to a set of people in various settings, including research labs, academic conferences, and university campuses. The sensors discover each other if they are in close physical proximity, and the resulting trace is a log of communication opportunities available among the participants over the course of the experiments. Each experiment includes tens or slightly more than a hundred participants.
- MIT Reality mining project [21] uses cell phones to collect encounter traces. Using the Bluetooth radio module, the software installed on phones keeps track of all Bluetooth capable devices in close proximity of the phone. Thus, it collects not only the encounter events among the phones, but also the encounter events of the phones and other "external" Bluetooth capable devices apart from the participating

phones. The trace collection lasted for about one academic year and involved about a hundred participants.

3.1.3 Other Approaches

The above two approaches represent the ends of a full spectrum. A combination of these approaches may be used. For example, in a study of end users watching video from popular sites (e.g., Youtube.com), the researchers obtain statistics for the most popular video clips from the website itself, use web crawler to understand the linkage between video clips, and sniff HTTP packets to analyze each video download transaction made by users [6].

On a different note, there are also efforts in the research community to come up with realistic mobility models that capture the mobility patterns observed within real traces. Major types of realistic mobility models include the following two major types (among others). In the "location-preference based" models, location preferences and time-variant, periodic user behavior are two important features to capture. Prominent examples of such mobility models include the TVC model [12, 22] and the ORBIT [5] model. Another class of mobility models uses the argument that the influence from other users (social ties) is the main driving force under the movement decisions made by users [19]. Mobility models based on social network theory is then generated.

3.2 Representation of Behavior

After the trace is collected, it is important to distill the raw data format (usually presented by [time, event] pair as the events of interest occur in the environment) and have a more precise representation towards the behavior to be leveraged later. This step is of crucial importance, since a proper selection of behavior representation makes it much easier to handle the succinct summary than dealing with the raw trace, and helps to represent information in a way that suppresses noise and focuses on the subject behavior of interest. Typically, it is an iterative research process to decide the best possible representation to shed the best light on the raw data. Each selected representation needs to be assessed for its effectiveness, as described in Sect. 3.3 below. More importantly, the selected representation needs to be proven useful for the target applications.

The selection of behavior representation is application dependent. As far as routing is concerned, it is usually the mobility (i.e., patterns of location visits) of users and the opportunities of communication (i.e., physical proximity of users, often referred to as "encounters") that are considered to be the most important behavior to capture, since they determine the possible paths of message flow in the network. In the following, we use examples from the literate to show various possible ways to summarize and represent user behavior from raw traces.

3.2.1 Location Preference: Association-Time Vector

In the most common format of WLAN traces, user associations to access points are captured as individual events. To understand the mobility pattern of a given user, however, it makes more sense to follow all events from the user and put them in a summarized form. The most popular way to do so is to sum up the total time the user spends at a given AP (across multiple association events at this AP). By listing this sum of association time at each AP in a vector form, it captures the mobility preference of the user (i.e., a large entry in the vector indicates the corresponding AP is heavily visited by the user). Note that this summary can be obtained by either centrally processing the WLAN trace, or by each user individually observing the access points it associates with.

This is one of the simplest and widely used forms to represent user mobility preference. It has been shown that on university campuses or research buildings, user mobility preference is highly skewed, meaning a given user only visits a small set of access points among all available ones, and spends most of online time at these access points (e.g., typically, users spend about 95 % of their online time at as few as five access points) [9].

3.2.2 Location Preference: Association Matrix

While summing online time spent at each access point provides a first-order representation for user mobility preferences, it does not capture potential variations of mobility preference of a given user at different time slots. For example, it is highly probable that the same student attends different classes thus visits different parts of a campus on different days of the week. It is also very likely that a person shows different mobility patterns during weekdays versus weekends. A study of mobility traces from university campuses reveals that most users display "multi-modal" mobility [10].

Therefore, a more granular representation than a single vector is required. One proposal that captures user mobility with multiple vectors is illustrated in Fig. 3, known as the association matrix. In the association matrix, each row is an association time vector for a different time slot. That is, each row vector captures the percentage of online time the user spent at various locations during this time slot. The time slot can be chosen based on the granularity desired for the representation. Typically, each row is used to represent a different day, since daily cycle is the most natural boundary in human behavior.

While the association matrix captures user mobility preference and its variation across time slots, it is a much more lengthy representation. Fortunately, researchers have discovered that there exists high repetition in human behavior, and thus the association matrix can be efficiently compressed. Using a standard technique for matrix summarization, singular value decomposition, it is shown that only a handful of vectors are sufficient to capture most of the variation in association matrix (e.g., five vectors are enough to capture typically more than 90 % of power in most user's association matrix), based on the study of WLAN traces from university campuses

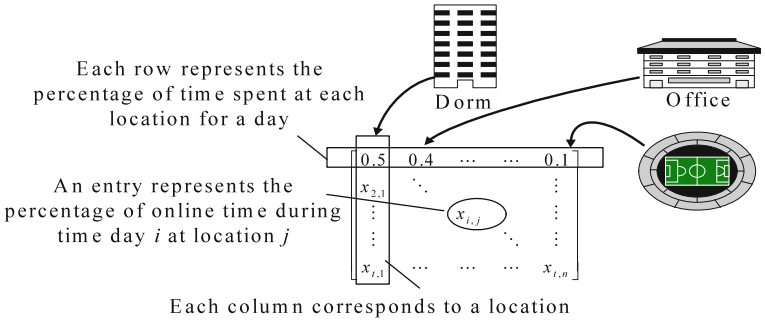


Fig. 3 Illustration of association matrix

(Dartmouth College and University of Southern California) [10]. We refer to these Eigen-vectors as Eigen-behavior for the users [4, 10].

The strength of the Eigen-behavior representation stems from its autonomous and succinct nature. Each node can generate its Eigen-behavior autonomously without involving other nodes, and this representation is small enough for easy exchange between nodes to facilitate behavior-aware routing protocols. We will further display this in the next section.

3.2.3 Node Centrality

Another approach to capture user behavior is by graph analysis. Consider each mobile device in the network as a node, and an edge between two nodes and its weight represent the likelihood of the mobile devices meeting each other, the whole mobile network can be summarized as a graph.

Now, to route packets across this graph, it is crucial to understand that nodes play different roles in the graph. Some of the nodes are more important in terms of connecting people in the network as they are placed at the center of the graph, through whom the connections of otherwise disconnected nodes are made possible. This property can be discovered by calculating the centrality of each node in the network. This metric can then be used as a single figure to measure the relative importance of nodes in terms of bridging the network for others. While traditional definition and calculation of betweenness centrality requires complete knowledge of the graph, which cannot be done by individual nodes, a modified definition of egocentric betweenness centrality [17] can be derived individually by each node with local, partial knowledge of the network and is thus useful for decentralized applications such as routing.

3.2.4 Delivery Probability Vector

A major event of interest to facilitate behavior-aware routing is the encounter among mobile nodes (i.e., two nodes move within radio communication ranges). When an encounter event happens, the two involved nodes have an opportunity to exchange messages.

One representation to capture the likelihood of nodes being suitable candidate for messages targeted for a given destination is the delivery probability vector, first introduced in one of the earliest behavior-aware routing protocol PRoPHET [16, 20]. The delivery probability vector contains as many entries as the number of total nodes in the network. If node A encounters with node B often, then A is a good candidate for messages targeted for B. This is reflected by increasing the entry for node B in node A's delivery probability vector each time they encounter. The delivery probability vector also considers transitivity, that is, if A meets with B and B meets with C often, then C can also be considered a suitable candidate for messages for A. Finally, if two nodes have not encountered for an extended period of time, then they are not as good a candidate to deliver messages for each other as before. Thus the delivery probability vector ages with a multiplication factor less than 1 periodically as well.

3.3 Validation of User Behavior

After the representation of user behavior is obtained, how do we validate that the particular summary is chosen in a meaningful way? With the plethora of wireless network user traces available today, and the rich information available from these data sets, it is possible to come up with almost infinitely many ways to represent user behavior. However, after one comes up with the representation that seems to be the most relevant to the task at hands, it is important to validate the representation to see if it is a meaningful way to look at the data.

Firstly, we want to check the stability of the representation—if the chosen representation truly reflects an intrinsic property of the users, it is unlikely for the representation to change drastically for a given user with respect to time. It is thus important to observe how the chosen representation for a user varies across time, and whether such variation can be justified. If the representation fluctuates a lot without reasonable justification, then perhaps the representation does not capture an intrinsic user behavior.

Secondly, one very important condition to consider is whether the representation leads to a well-separated classification of users. It is typical that when a social behavior is meaningfully captured, users show different patterns based on the representation. Leveraging these natural differences among users, a classification can be drawn with well-defined boundaries. While users in the same group are highly similar to each other based on the given representation, users in different groups are drastically different. This result can be compared against putting users randomly in groups and the classification based on the behavior representation should show much

more significance. If this cannot be achieved, perhaps the representation is not doing a good job discerning users, and hence is not a good candidate when we want to leverage users with different behaviors for routing purpose.

While validating the representation, it is also worthwhile to look at multiple sources of user traces and dig into the similarity and difference in user behavior. Many times, population with similar properties (e.g., students from similar university campuses) would display similar behavior trends. The qualitative similarity can often be striking. In contrast, different sub-user populations (e.g., laptop v.s. PDA) may display different behavior trends. By analyzing various populations or sub-groups within a population and trying to explain the similarity or difference discovered, one can understand deeper if the representation has captured user behavior meaningfully.

In the following subsections we further elaborate these points with examples from relevant research papers.

3.3.1 Time Stability of the Representation

Firstly, if the user behavior representation really captures an intrinsic aspect of user behavior, it is expected the trend should be stable (i.e., not varying randomly) for the same user. After all, one major reason we rely on user behavior representation for routing is that we expect the past behavior to be a good predictor for future behavior, and thus we can count on the persistence of user behavior to make intelligent routing decisions. If this is not the case, there is not much use to base routing decisions on user behavior. Thus, it is crucial to verify the representation has at least some stability for a meaningful time horizon for the application.

For example, in [11], based on WLAN traces collected from university campuses, the representation of Eigen-behavior of a given user is compared against its future behavior. As illustrated in Fig. 4, at two different time points T days apart, the same user's association matrix is created using the trailing d days of traces, and the corresponding Eigen-behavior vectors are calculated. The similarity score, as defined in [10], is then calculated between the two resulting sets of Eigen-behavior. The results are shown in Fig. 5. We can observe a few interesting properties from the figure. (1) the similarity score remains high for the same user for a long time into the future. When we consider $T = 35$ days (5 weeks) apart, the mobility profiles from the same user still show high similarity, with values higher than 0.6. This implies the current association matrix and the derived Eigen-behavior is a reliable predictor of user behavior into the future. (2) The amount of history used does not influence the result too much when the considered T is large enough to avoid overlaps in the used mobility history (i.e., when $T > d$).

The stability gives us the confidence that once the Eigen-behavior of a user is derived, it remains similar for a period of time. This period of time is long enough for the purpose of message delivery in mobile networks so that the Eigen-behavior can be leveraged as guidelines based on which routing decisions are made. Even if two nodes do not encounter frequently to update their Eigen-behavior with each other, the historical information is still meaningful and useable. It is such a stability that

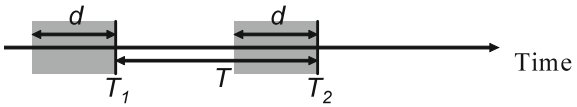


Fig. 4 Illustration: Compare the Eigen-behavior vectors obtained from trailing d-days trace at two points T days apart

Fig. 5 Average similarity metric of the Eigen-behavior from the same user at T days apart

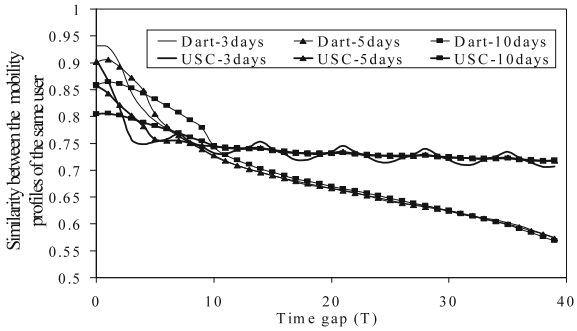


Table 2 Percentage of power captured by the most important Eigen-behavior from each user group, from two WLAN traces

	For its own group	For other group
USC	0.779	0.005
Dartmouth	0.727	0.004

creates a structure in social behavior that can be explored by routing protocols. While we show only an example with Eigen-behavior, similar stability analysis should be carried out as a first step to examine if a given metric can be considered as a reliable indicator for behavior-aware routing.

3.3.2 Significance of User Grouping

Another way to examine the validity of a representation is to compare whether a certain trend is likely to appear due to random chances with statistical analysis. If it is not likely, then the representation has captured something significant, and intrinsic to user behavior rather than due to chances. This is another important way to verify that the representation is useful.

One example of this verification is done to user grouping based on similar mobility trends in [10]. Once the grouping has been determined, we would like to verify if the mobility trend is indeed consistent and unique for each identified group. In order to determine the uniqueness, we take the most dominant Eigen-behavior from each user group and calculate how much power it captures for this user group and other user groups. From Table 2 we can easily see each of the top group Eigen-behavior

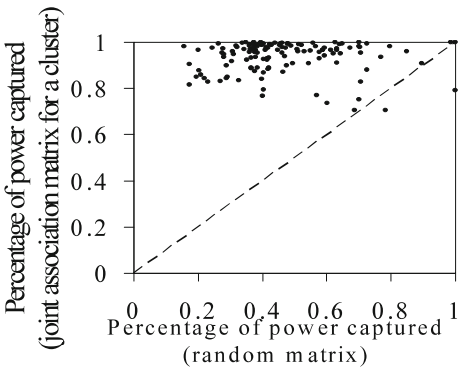


Fig. 6 Scatter graph: Cumulative power captured in top four Eigen-behavior vectors of random groups (X) and groups formed according to similarity in Eigen-behavior (Y)

captures significant more power within its own group than other groups. This verifies the uniqueness of identified groups in terms of user behavior. Each group collects nodes with behavior different from other groups.

In order to determine the consistency of behavior of users in the same group, we calculate the percentage of power captured by the top Eigen-behaviors from the whole group. The idea is, if the users in the group follow a coherent behavioral trend, the percentage of power captured by the top Eigen-behavior vectors should be high. On the other hand, if users with different trends are mistakenly put in one group, the percentage of power captured by its top Eigen-behavior vectors should be much lower. We then compare the cumulative power captured by the top four Eigen-behavior vectors of our groups with random groups of the same size (i.e., randomly pick the same number of users from the whole population) in scatter graphs, in Fig. 6. Clearly, most of the dots are well above the 45-degree line, indicating the users in the same group follow a much stronger coherent behavioral trend than randomly picked users, and therefore the user grouping is of statistical significance.

3.3.3 Cross-Validation: Experiment with Multiple Data Sources

One additional step to verify validity of a particular behavior representation is to cross validate with multiple data source. If a representation works well not only on one, but multiple data sets, we have better confidence that this is a genuine way to capture user behavior. Therefore, in many research work on this front [3, 9–11, 13], we see the results are reported for multiple data sets, while similar observations are made. This is not just a repetition. While at times the data sets are obtained from similar settings (e.g., traces from different university campuses), it is a very important step to prove that the behavior trend captured is not an artifact from a specific place, but at least a common behavioral trend from similar environments.

4 Behavior-Aware Routing Protocols

In the following sections we use studies in the literature as examples to present the state-of-art of user behavior data collection, representation, and summarization. We further discuss several studies that use these behavior representations for routing tasks.

Once we obtain a valid representation of user behavior, it can be leveraged for many applications. Routing is just one very important example. The fundamental argument for behavior-aware routing is that human behavior is repetitive. Therefore, based on the past history, if some event happened frequently, it is likely to happen again, and this trend can be leveraged to determine which nodes are better candidates to deliver messages for a target.

Typically, behavior-aware routing leverages a greedy, gradient-ascend approach. While the actual metric and protocol operation details differ, most of them use a similar principle: Starting from the source node, each node looks for other nodes that are better candidates (i.e., more likely to deliver the message successfully) and in turn sends the message to these promising intermediate node(s). These nodes then operate similarly to find even better candidates for the message, and this operation continues until the message is delivered. Each protocol differs from others by the metrics they use, the way these metrics are collected and maintained, the protocol operation details such as using single or multiple message copies, amount of control messages needed for protocol operation, recovery mechanism from faults, or termination conditions. This general form of behavior-aware routing protocol is illustrated in Fig. 7.

In this section we introduce some well-known works from this very new research area to provide examples of behavior-aware routing protocols.

4.1 PRoPHET

PRoPHET [16] is one of the earliest work that introduces the notion of behavior aware message routing. In PRoPHET, a delivery probability vector is maintained by each node, detailing the probability that it is a good candidate to deliver messages to all the nodes in the network. Consider a given node A in the network. If it encounters with node B repeatedly, then the delivery probability for node B should be increased in node A's vector. Also, if node B meets with node C, D frequently, by transitivity, node A should also increases its delivery probability for node C, D as it can get to these nodes through B. Finally, all entries in the vector slowly decay with time as an extended time period absent of encounter events should be reflected by reducing the delivery probability. When nodes encounter with each other, they exchange the encounter probability vectors in order to update each other about encounter transitivity.

When it comes to message delivery, each node simply looks out for other node which has a higher delivery probability for the message destination node than itself

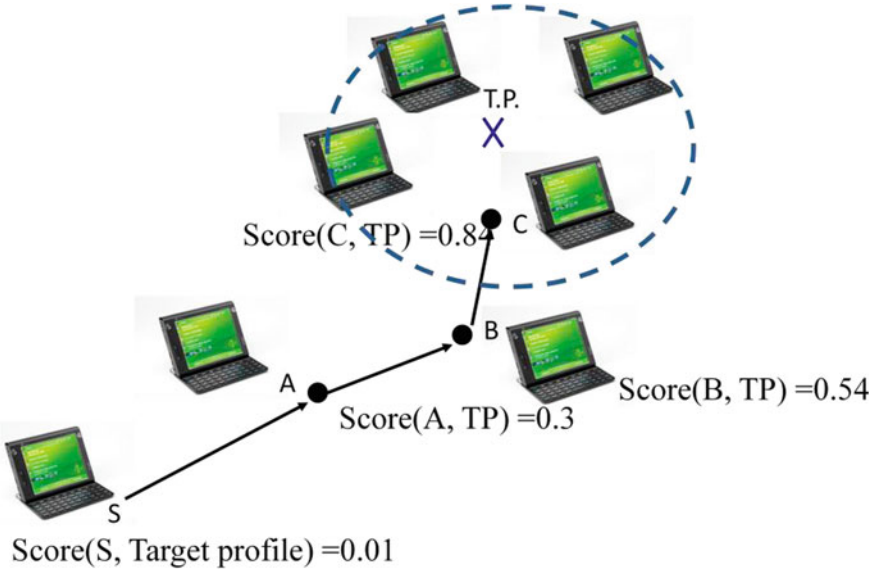


Fig. 7 Conceptual illustration of gradient-ascend routing in behavior space—Each node calculates a score based on its own behavior and the target profile. Based on this score, the message is transmitted towards node ranked higher, progressing towards the target profile

as the next hop. The message follows an increasing gradient in delivery probability to reach the final destination.

4.2 Mobility-Pattern-Space Routing

Another early work in behavior-aware routing, in [15] the authors introduce the concept of MobySpace. Each node is represented by a single coordinate in this space, using the representation of association vector as introduced in Sect. 3.2.1. The protocol assumes that the association vector of the destination node is known to the sender, and it is used in the place of destination node ID for message delivery. A message follows a pure greedy ascend approach across the network. Current message owner attempts to find another node with an association vector more similar (i.e., shorter Euclidean distance) to the destination than itself and relay the message to this node. This process continues until the message reaches the destination node.

This work is among the first ones to look at repetitiveness in user behavior, in this case mobility, and leverage it for routing. It turns out performing much better than other simple strategies, such as random routing, hot potato routing, or waiting for the sender and destination nodes to meet directly, in terms of both delay and success rate.

This is one of the earliest work to leverage mobility pattern (a form of behavior) to represent the destination of a message.

4.3 Social Orbit

The notion of social orbit is proposed by Ghosh et al. in [5]. It is likely for an individual to have a bunch of frequently visited locations (known as "hubs" in their terminology), and a set of probable sequences by which the individual visits these locations (known as "social orbit" in their terminology). This social orbit exists at micro-level (e.g., buildings on a university campus) and at macro-level (e.g., cities in the nation). It is highly likely that an individual repeats her mobility pattern from one of the already known social orbits.

Once the social orbits are identified, they can be leveraged in a routing protocol. The authors propose a Sociological Orbit aware Location Approximation and Routing (SOLAR) protocol, which works as follows. In this protocol, if the source node already knows the social orbit of the destination node, the message is sent (through greedy geographic routing [14]) towards the geographical central point of all hubs of the destination node. Once the source and destination nodes connect with each other (i.e., exchange the first packet of a stream of packets), they include their updated social orbit and the current hub they are at within the packet header to inform each other with better location knowledge. Subsequent packets are then delivered using geographic routing to the location of the other party directly. Once two nodes exchange messages they become acquaintances and store the social orbit information of each other.

If the social orbit of the destination node is not yet known to the sender, then sender carries out a series of actions to discover the social orbit of the destination node. Query packets are sent from the sender to its acquaintances in order to check if one of the acquaintances knows about the destination's social orbit. Among all the acquaintances, a subset is chosen carefully to maximize the number of hubs covered by the queries while minimizing the number of queries sent (i.e., pick the acquaintance that goes to hubs that no one else goes first).

4.4 Mobile Social Network Routing

Social network properties can also be leveraged for routing protocol, as displayed by Daly et al. in [3]. In this work, instead of focusing on individual behavior properties, such as mobility pattern, the authors try to identify the role a node plays in the larger network of mobile nodes by quantifying several network properties. They choose (1) a node's betweenness centrality, (2) a node's social similarity to the destination node, and (3) a node's tie strength relationship with the destination node, as the metrics to pick promising next hop for message forwarding.

Betweenness centrality is a measure of the extent to which a node has control over information flowing between others. While the correct number of betweenness centrality cannot be obtained without the knowledge of the whole network topology, each node in the network can however use only its local view to derive a measure of egocentric betweenness, which has the same ranking as the sociocentric betweenness (the one obtained from complete network knowledge). The strength of ties between mobile nodes can be calculated based on the frequency, duration, and recency of their mutual encounter events. And finally, the social similarity of two nodes can be estimated by the size of the group of common nodes they both encounter with.

The authors then devise a utility function to evaluate all possible intermediate nodes for their suitability of message forwarding, incorporating all three metrics mentioned above. For a given message, if only one copy is allowed in the network, it is transmitted when another intermediate node with higher utility is encountered by the current message holder. If more than one copy of a message is allowed, when the current message holder meets a new node, they split the number of copies of message each of them is responsible to distribute later, according to the ratio of their utility function.

In another work by Hui et al. social network structure is also considered in a routing protocol named Bubble Rap [13]. The authors consider two important factors in human network, community and popularity. They obtain the community structure from inter-node encounter traces using standard community detection algorithms, such as K-clique or weighted network analysis. Popularity of a node within its community and global popularity are then measured by betweenness centrality. One key observation from the authors is that a globally popular node may not be the best forwarder for messages destined at a given community if this node does not belong to the community. Therefore, the Bubble Rap protocol works in stages. First, a message follows the global popularity metric to reach a globally popular node. But as the message progresses in the network, at the later stage, it is the local popularity within the destination community that better determines which nodes are the most useful to deliver the message. Using the knowledge of the community of the destination nodes, the local popularity metric then can be used for better forwarding decisions.

In the work by Costa et al. [1], a publisher-subscriber model is integrated with message delivery in social network. Their fundamental assumption is that similar interest implies higher probability of co-location (i.e., birds of the same feather flock together). This assumption is applicable in some cases to facilitate behavior-aware message delivery. Based on the assumption, the authors device a utility function that considers two factors (1) whether a given node co-locate with any nodes with certain interest i, and (2) if a given node has a lot of connectivity changes (i.e., addition and removal of neighbors), to device a utility function for this node's utility to deliver messages for interest i. This work provides some indirection between message destination IDs and message destination properties (e.g., interest).

The success of utilizing social network metrics for routing decision indicates that not only individual behavior, but also social behavior, displays a repetitive pattern in our daily life. A node that is central to a network in the past is likely to remain

central, and a strong tie among two nodes is likely to persist. Therefore, inter-node relationship could be leveraged to build efficient behavior-aware routing protocols as well.

4.5 Small-World and CSI

Most of the previously discussed work use either individual behavior patterns (e.g., encounter or mobility history) or social behavior patterns (e.g., betweenness, community) for message forwarding decisions. However, it is yet unclear how these two types of behavior patterns are related to each other.

In the work by Hsu et al. [11], it is found that individual nodes display stability in their mobility profiles (as we discuss in Sect. 3.3.1). More interestingly, there is a linkage between the similarity of this mobility profile and the social relationship between nodes. If two nodes are similar in their mobility profiles, they are likely to encounter, and encounter with much longer time duration (as shown in Figs. 8 and 9). On the contrary, if two nodes show very dissimilar mobility profiles, they are unlikely to encounter, but the encounter probability is not zero (Fig. 8). More interestingly, when we compare the sets of nodes encountered by two given nodes, we discover that if the two nodes are dissimilar in mobility profiles, the sets of nodes they encounter directly are very different (Fig. 10). Thus, these rare and random encounters between dissimilar nodes are very important events in a mobile network for message routing. They bring the distance across the network much smaller, by introducing an opportunity to exchange messages among group of nodes which are otherwise unlikely to meet. A social network is typically a Small-world network [9]. The authors argue that the cliques in the social network are formed by people with similar mobility profile and thus encounter often. The random links that bring the network distance shorter are formed by the random encounter events between dissimilar nodes.

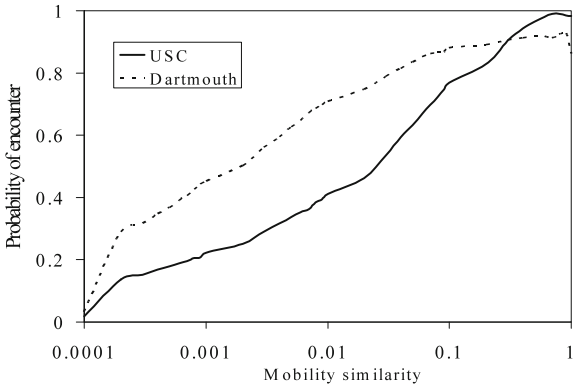


Fig. 8 Mobility similarity leads to higher probability of encounter

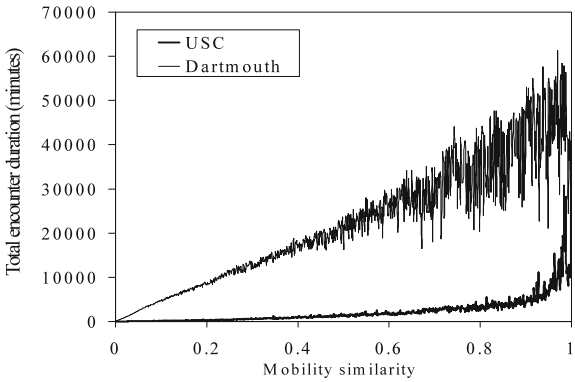


Fig. 9 Mobility similarity leads to longer duration of encounter

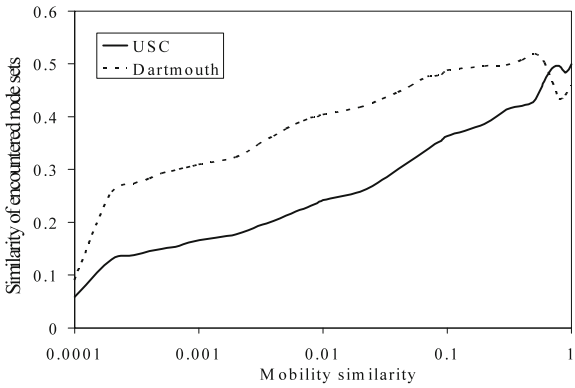


Fig. 10 Mobility similarity leads to similarity in the sets of nodes encountered

With this property in mind, the authors further design CSI protocol for message dissemination in social networks. The proposed CSI protocol has two modes: CSI:Target (CSI:T) mode and CSI:Dissemination (CSI:D) mode. The idea behind CSI:T is somewhat similar to mobility space routing discussed in Sect. 4.2. Given a target mobility profile, intermediate nodes send the message towards another node with more similar mobility profile to the target profile, thus improving the chance to meet with target nodes. This is illustrated in Fig. 7. CSI:T can be used when the mobility profile can be used to represent the target of a message.

CSI:D mode provides a more generic protocol suitable for message delivery towards target profiles that cannot be represented by a mobility profile. One example is to reach people who like movies on a university campus. If there are no movie theaters on the campus, the measured mobility profile cannot be used to infer such interest. This challenge is pointed out in Sect. 2 (Fig. 1)—nodes that are similar in the interest space may not be similar in mobility behavior space, thus the gradient ascend approach does not work. There appears to be little insight provided by the similarities between the nodal mobility profile to guide message propagation, as the intended receivers in this case may be scattered in the behavior space, and the relationship

between the target profile and the mobility profile cannot be easily quantified. However, the authors argue otherwise, suggesting that mobility profile can be of use even if the target of message delivery has nothing to do with mobility itself.

The objective of CSI:D mode is to introduce a small number of message copies transmitted and stored in the network intelligently, yet make it possible for most nodes to get a copy quickly, if they are the intended receivers. CSI:D leverages the discovery made earlier on the relationship between nodal encounter patterns and the similarity between mobility profiles. That is, nodes with dissimilar mobility profiles are more likely to encounter with different sets of nodes, while nodes with similar mobility profiles tend to meet often (i.e., belonging to the same social clique). Therefore, when the objective is to reach most of the nodes in the network with lesser message replication, it is wise to keep only one copy of the message in each social clique, and make a copy only if a message holder encounters with a node that displays dissimilar mobility profile from all known message holders. In this regard, CSI:D protocol attempts to spread copies of the message out to the network intelligently, leveraging only the rare but important encounter events in social network—the encounter between different social cliques, or between nodes with dissimilar mobility profiles. With this strategy, the authors show that the mobility profile can still be useful to guide message delivery, even if the target of message has nothing to do with mobility patterns.

5 Discussions

5.1 User Privacy

Collecting user behavior and using such behavior for message delivery certainly raises privacy concerns. While the full picture of privacy implication of behavior-aware routing protocols and the necessary means to protect user privacy are still under research, the following comments can be made with respect to user privacy in behavior-aware routing.

Firstly, While the research of behavior-aware routing uses the complete information from user traces as a mean to understand user behavior, full access to such traces is not necessary for the routing protocols given in Sect. 4 to operate. Specifically, each participating node does not need complete knowledge of all other nodes in the network (in fact, this is an important requirement for routing protocol in mobile network—it has to work in a decentralized fashion). Each node merely collects its own behavior and exchanges with nodes it communicates with. Also, there is no "service provider" responsible for setting up the network and marshaling message propagation. Thus, no one has the ability to track or store the behavior data of users in the network. Individual users own their behavior data, and can potentially control how much to share with other people.

Secondly, the behavior summarization step helps to reduce the granularity of information made available to other users. Using mobility pattern as an example, the raw event (I am at location X from time T1 to T2) is never shared. Instead, a summarized mobility preference vector (my most frequent locations are X, Y, Z, and I spend 50, 30, 15 % of time, respectively, at these locations) is shared. Furthermore, the summarization step may in fact remove the actual semantic of trace in some cases. All that the protocol cares is if two users are similar (e.g., visiting similar location sets), and exactly which locations they are is not really important.

Finally, users with privacy concerns may opt out of the protocol, and they can do so without severely impacting the network performance. One interesting property for social network is its robustness—it is show in [9] that the network of mobile nodes formed by encounters is robust to node removal. Even if 20–30 % of nodes choose to not participate in message routing at all, the messages still find alternative ways through the network.

5.2 Relationship to the Current Internet and Alternative Network Structures

In this section, we briefly discuss the relationship of the behavior-aware routing protocol to the current Internet, and several other proposed alternative network structures in literature.

Until now, most communication paradigms and primitives in computer networks have been identity-centered. Messages are destined to a machine identity (e.g., unicast, multicast or broadcast IP addresses) when they are sent across the network. Thus, in traditional network protocols, searching and addressing services need to be maintained for the purpose of looking up the information provider's machine identity (i.e., use Google for the library website and use DNS to find its IP address), and the parties are then connected using identity-centered protocols. This is illustrated in Fig. 11a, where well-known look-up services become an essential part in all communication. However, communication occurs in many cases for a purpose different than reaching a particular machine ID (e.g., I want to find out information about library hours, whoever actually provides the information is irrelevant to me). Social behavior-aware routing, on the contrary, has the potential to provide a new paradigm where the behavior itself is used to guide message delivery. There is potentially more than one specific receiver identity in many scenarios. Instead, whoever matches the target social behavior of the message are all potential receivers. The network protocol itself collects, maintains, and routes messages based on behavioral properties of participating nodes in the network, in a distributed fashion. This is illustrated in Fig. 11b, where the sender specifies the target behavioral profile in the header of the message and injects it into the network. While this paradigm deviates from traditional networking, it is not meant to replace existing paradigms but to augment them. We do believe that in many contexts, there is really no hard requirement to reach specific

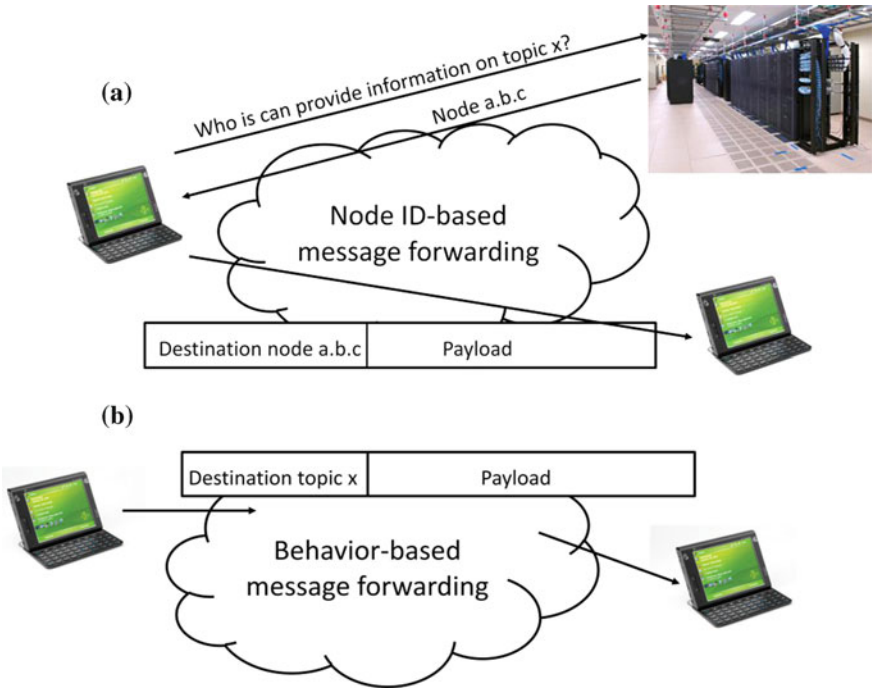


Fig. 11 Traditional ID-based versus behavior-based message forwarding schemes, **a** Node ID-based message forwarding, **b** Behavior-based message forwarding

node(s). Instead, matching communication parties based on the desired behavior patterns may be the most efficient, robust, and application-centric paradigm in mobile networks in many cases.

6 Conclusion

With the advances of mobile computing and communication devices, it is envisioned that each person will possess personal devices exclusively used by the owner. This emerging trend makes it possible to leverage these devices as sensors for human behavior to collect usage data from the owners. While many applications are possible with this collected data, in this chapter we focused on its usage in behavior-aware mobile social networks. The traces are used to understand different roles played by each individual from networking perspective, and leverage this understanding to deliver message intelligently by selecting appropriate forwarders. This new research area, which we describe as social behavior-aware routing, is actively under development.

In order to successfully design a social behavior-aware routing protocol, it is imperative to collect realistic user data, summarize and represent the user data in a suitable form for the application, and examine the validity (stability and significance) of the representation. It is also beneficial if the chosen representation can be calculated by each node with its local knowledge, enabling decentralized application.

While most of the behavior-aware protocols in the current literature still use machine IDs as the final destination of messages with behavior awareness being used as a mean to facilitate this goal, a new possibility with behavior-aware routing is to use the actual behavior, instead of a machine ID, as the destination. We argue that this form of routing is actually more natural in many applications such as targeted advertisement or topic-based discussions. However, the linkage between general interest and user mobility (the main facilitating factor of communication in mobile networks) is to be further studied and understood.

References

1. P. Costa, C. Mascolo, M. Musolesi, G. Picco, Socially-aware routing for publish-subscribe in delay-tolerant mobile ad hoc networks. IEEE J. Sel. Area Commun. **26**(5), 748–760 (2008)
2. CRAWDAD: A Community Resource for Archiving Wireless Data At Dartmouth, http://crawdad.cs.dartmouth.edu/index.php
3. E. M. Daly, M. Haahr, Social network analysis for information flow in disconnected delay-tolerant MANETs. IEEE Trans. Mobile Comput. **8**(5), 606–621 (2009)
4. N. Eagle, A. Pentland, Eigenbehaviors: identifying structure in routine. Behav. Ecol. Sociobiol. **63**(7), 1057–1066 (2009)
5. J. Ghosh, S. J. Philip, C. Qiao, Sociological orbit aware location approximation and routing (SOLAR) in MANET. ELSEVIER Ad Hoc Netw. J. **5**(2), 189–209 (2007)
6. P. Gill, M. Arlitt, Z. Li, and A. Mahanti, YouTube traffic characterization: a view from the edge, in *Proceedings of Internet Measurement Conference (IMC)*, Oct 2007
7. Haggle Project, http://www.haggleproject.org/
8. T. Henderson, D. Kotz, I. Abyzov, The changing usage of a mature campus-wide wireless network, in *Proceedings of ACM MobiCom 2004*, Sept 2004
9. W. Hsu and A. Helmy, On nodal encounter patterns in wireless LAN traces. IEEE Trans. Mobile Comput. **9**(11), 1563–1577 (2010)
10. W. Hsu, D. Dutta, and A. Helmy, Mining behavioral groups based on usage data in large wireless LANs. IEEE Trans. Mobile Comput. (accepted and to appear)
11. W. Hsu, D. Dutta, A. Helmy, CSI: a paradigm for behavior-oriented profile-cast services in mobile networks. Elsevier Ad Hoc Netw. (accepted and to appear)
12. W. Hsu, T. Spyropoulos, K. Psounis, A. Helmy, Modeling spatial and temporal dependencies of user mobility in wireless mobile networks. IEEE/ACM Trans. Netw. **17**(5), 1564–1577 (2009)
13. P. Hui, J. Crowcroft, E. Yoneki, Bubble rap: social-based forwarding in delay tolerant networks, in *Proceedings of MobiHoc*, May 2008
14. B. Karp, H. Kung, GPSR: greedy perimeter stateless routing for wireless networks, in *Proceedings of ACM MobiCom*, Aug 2000
15. J. Leguay, T. Friedman, V. Conan, Evaluating mobility pattern space routing for DTNs, in *Proceedings of IEEE INFOCOM*, April 2006
16. A. Lindgren, A. Doria, O. Schelen, Probabilistic routing in in-termittently connected networks. ACM SIGMOBILE Mobile Comput. Commun. Rev. **7**(3), 19–20 (2003)
17. P.V. Marsden, Egocentric and Sociocentric Measures of Network Centrality. Soc. Netw. **24**(4), 407–422 (2002)

18. MobiLib: Community-wide Library of Mobility and Wireless Networks Measurements, http://nile.cise.ufl.edu/MobiLib
19. M. Musolesi, C. Mascolo, A community based mobility model for ad hoc network research, in *Proceedings of the Second International Workshop on Multi-hop Ad Hoc Networks (REALMAN)*, May 2006
20. PRoPHET IETF Draft, http://tools.ietf.org/html/draft-irtf-dtnrg-prophet-09
21. Reality Mining Project, http://reality.media.mit.edu/
22. Time-Variant Community Mobility Model, http://nile.cise.ufl.edu/TVC_model/

Chapter 18
Emerging Applications of Wireless Sensing in Entertainment, Arts and Culture

Jeffrey A. Burke

Abstract This chapter provides an overview of cultural applications of wireless sensing systems from four perspectives: the "Internet of Things", the "Smart Grid", "participatory sensing" on mobile phones, and an event-based point of view. The challenges and unique requirements of these applications are examined, and future opportunities for research are suggested in three technical areas: Machine Learning, Networking and Privacy. The need for more advanced authoring tools—enabling creators of cultural applications with less technical backgrounds to develop complex systems that use wireless sensing—also motivates work on these three technical topics.

1 Introduction

This chapter explores the current and emerging roles of wireless sensing (and related examples in embedded sensing) in *cultural applications*—entertainment, arts and culture—where it is offering new opportunities for human creativity and experience. Three related technological movements are reviewed: the *Internet of Things*, which focuses on *IP-enabled objects*; the *Smart Grid*, which involves *network-connected infrastructure and utilities* (water, power, etc.); and *participatory sensing*, which concerns *personal wireless devices* in the consumer market and related research in *sensing* using these devices. Each movement provides a specific perspective on the potential for culture-related uses of wireless sensing.

While these three areas share common technology and overlapping research agendas, their visions intersect with the cultural sphere in different and informative ways. Also described is a fourth, less pervasive perspective—that of *events*—represented most formally through live performance, but also applicable to a variety of human

J. A. Burke (✉)
University of California, California, LA, USA
e-mail: jburke@ucla.edu

H. M. Ammari (ed.), *The Art of Wireless Sensor Networks*,
Signals and Communication Technology, DOI: 10.1007/978-3-642-40066-7_18,

encounters. This perspective poses complementary challenges to the creators of wireless sensing systems. The chapter concludes by describing research topics in wireless sensing that hold promise for empowering cultural users to author more sophisticated, ubiquitous applications.

Cultural applications, as described here, include the arts, entertainment, and projects of civic and cultural engagement. Such applications have aesthetic, ludic, experiential and social objectives that extend beyond simply supporting communication or efficient task completion using technology. They relate to experiences with processes that strive to engage its participants, who are willingly committing their free time to be engaged; they are not necessarily supposed to be quick or seem efficient. Compared to commercial technology, cultural applications have a less utilitarian perspective on their employment of technology and involve decisions that may more frequently and more explicitly challenge (and delight) their users.

The approach of this chapter emerges from 15 years of experimentation at the University of California, Los Angeles. The research explores the roles of new technologies in cultural applications, and is part of a larger, ongoing movement by many cultural producers and critics. Sensing has played an important role in this area, connecting the physical world occupied by humans with the digital media and tools that are now fundamental to many applications within the arts, entertainment and culture.

Sensing is part of the changing sphere of cultural production. Other elements of the expanding palette for creators include: more sophisticated tools for capturing, distributing, and using digital media; new tools for infusing data into built environments through changes in lighting, material characteristics, sound, and other media; expanded modes of storytelling using nonlinear, locative, and database-driven structures; and new paradigms for interconnecting components, such as information-centric networking. It is within systems of these components that wireless sensing is considered.

2 Perspectives

2.1 From the Internet of Things to the Internet of Places and Experiences

Mark Weiser's early vision for ubiquitous computing has motivated research and consumer product development for two decades [30]. One recent evolution of the ubicomp vision has been the *Internet of Things* concept, which envisions almost every device and many objects as IP-enabled (as illustrated in Fig. 1) and, to varying extents, context-aware [10]. Within this vision, wireless sensing is integrated into Internet-connected objects, as well as deployed as independent, Internet-connected infrastructure.

Integrating Internet-enabled *things* into cultural experiences can be pursued object-by-object. For example, the ways to integrate IP-enabled "things" into so-

Fig. 1 Artist's rendering of the *Internet of Things* concept

cial networks and other existing networked systems can be considered as needed. Perhaps reframing the discourse in this research area to consider *places* and *experiences*, instead of *things*, would a significant opportunity for the cultural sector. This terminology better represents the design challenges faced by those who make the "things" for people to use in their everyday lives.

In the Internet of Things, sensing is used both as inputs for direct HCI (*human-computer interaction* or human-device interaction) and for *observations* that do not involve direct interaction with a user. (These two modes exist within the other perspectives treated in this chapter as well.) Creators of cultural applications have explored both modes of operation; they have extended the notion of network-supported interaction and interconnection of everyday devices beyond task support to other human pursuits, including creating emotionally and aesthetically pleasing objects and engaging with the surrounding political, cultural and environmental context. In these cases, focuses on experience and place take precedence over any one object.

A significant (and representative) early exploration of HCI opportunities within the Internet of Things (before it was known as such) was Hiroshi Ishii's *Tangible Bits* [15]. Ishii and the MIT Tangible Media Group explored the idea of Tangible User Interfaces (TUIs) that "provide physical form to digital information and computation." They anticipated network-connected objects used for accessing information and often using external computer vision to create interactions, in addition to other sensing mechanisms. Not just everyday objects, these "objects of wonderment" (a term later coined by Paulos [23]) would encourage their users to think differently

Fig. 2 *IO brush* from the tangible media group at MIT, a cultural context example of future human-computer interaction possible in an *Internet of Things*. (Ryokai, Marti, and Ishii 2004)

about the possible connections between the digital and physical worlds. The Tangible Media Group's projects, such as the *IO Brush* in Fig. 2, emphasized shifts in user experience—made possible through technology embedded in everyday objects, not only the connection of those objects to digital systems.

The observational model for the use of sensing devices within the cultural realm can be seen in the work of artists, such as Stanza, who, in gallery exhibitions, sonifies and visualizes data from wireless sensors placed in the urban environment.[1] For example, his project, *Capacities: Life in The Emergent City* uses wireless networked sensors to capture temperature, light, pressure, noise and sound data from the city outside the gallery, and controls elements of a physical artwork in real-time based on this information. The project uses forty mote-based sensor packages as data sources. It is one of twelve such works created by the artist between 2004 and 2012.

3 Challenges and Opportunities

Designing complete experiences and physical environments that incorporate Internet-connected, sensor-enabled objects poses substantial technical challenges, as discussed in other chapters of this book. In cultural applications, aesthetic and usability requirements are paramount, even in experimental work, and often drive technical

[1] http://www.stanza.co.uk

requirements further than in the scientific fields that have motivated early work in sensing [4].

Regardless of experimentation, the complexities of wireless radios and mesh networking of devices had limited the creation of ecosystems of wireless sensors that would enable these types of experiences. As these challenges have been addressed and the technologies further commercialized, the Internet of Things vision has become more viable, as have extensions into everyday life that move beyond everyday objects.

The emergence of "maker" culture (Economist 2011) has brought significant experimentation into the cultural sector. Low cost and open source hardware has been designed to support experimentation with wireless sensing, including the Arduino[2] microcontroller platform and XBee[3] radios. Until recently, many experiments were networked through application-specific gateways (if networked all). With increasing support for IP and even HTTP on small devices, as well as web services such as Pachube (now Cosm[4]) for sensor data aggregation, the Internet of Things vision is developing. The prevalence of HTTP connectivity, for accessing sensor data at the gateway and device level, has made the incorporation of sensor data more viable for cultural producers primarily familiar with web development.

Cosm, in particular, presents an intriguing interface to the Internet of Things. It allows a global database of live data streams to be accessed through a consistent interface, making data access both consistent and approachable to "makers", while also centralizing it within a single web-based aggregator.

Cultural applications of wireless sensing will benefit from two new possibilities enabled by the Internet of Things networking of sensors with the commodity Internet: integration with media and federation of many devices. IP integration will enable the creation of Internet of Things (IoT) applications connected to the mainstream media world of high definition content and interactive environments—integrating capture, processing and dissemination of high-bitrate digital media, sensing and control, distributed processing, and user interfaces in increasingly sophisticated and complex combinations, with scales and diversity far beyond today's applications.

At the same time, related trends in cultural production—e.g., social networking, cloud-hosted media, location-based experiences, and multiplayer online gaming—suggest that the IoT will not only be used for a collection of isolated device-centric experiences, but for the creation of multi-person, multi-site, multi-device experiences. Cultural producers, responsible not only for individual devices but entire experiences, will need approaches to wireless sensing designed to be integrated into experiences and places in a holistic way, not simply to create networked things.

The network of objects will become a fundamental part of how places and experiences are created—in the home, workplace, entertainment-focused environments and "third places", what Ray Oldenburg called the coffee shops, public spaces, and

[2] http://www.arduino.cc

[3] http://www.digi.com/xbee

[4] https://cosm.com

other locations where we spend our time outside of home and work. And they will also be linked to the emerging world of Internet-supported media distribution.

The interweaving of wireless sensing systems (and Internet-enabled objects that incorporate them) with media presents engineering challenges—bridging the various networks involved, each with different power, bandwidth, and security requirements. It also presents more fundamental challenges—how to conceive media-based experiences that can be adjusted based on sensor data, and how to process sensor data in a way that generates aggregated values or features that are relevant to particular domains of cultural authorship. Network-enabled objects for public manipulation in a science museum, for example, require features and sampling rates to support their use as human-computer interfaces in that context, whereas a system of sensor-enabled objects designed to observe and illustrate patterns of crowd flow in a public park would have different needs. The Internet of Things perspective, in particular, suggests that key research challenges emerge *after* the vision of interconnectivity for a single object or device is achieved—a familiar but still pressing challenge for creators of wireless sensing systems, recast in the context of cultural production. To enable *authoring* of experiences and places built up from IP-connected devices that include sensing components is to drive a host of new innovations. (The last section of this chapter addresses some of the research challenges generated by cultural uses of wireless sensing, as articulated by the description of each technological perspective).

3.1 The Smart Grid as a New Host for Cultural Production

In contrast to the Internet of Things vision, which emphasizes Internet-connected objects and devices, the *Smart Grid* and "Intelligent Building" movements emphasize the instrumentation and interconnection of the entire built environment, including infrastructure and buildings. And recently, this has taken into consideration commodity networking and web interfaces for management and control. (See Fig. 4.) The emphasis is on taking the existing electrical "grid" and other similar resources, and providing highly granular instrumentation to support sustainability, energy management, and increased security across a variety of scales—from the desktop to the power plant [2], as illustrated in Fig. 3. In this context, wireless sensing is often employed for instrumentation, where wired connectivity is infeasible or the devices to be instrumented are mobile. Sensing as a whole is primarily a mechanism for observation, not *human-computer interaction*, which is done via web-based services and other graphical interfaces.

While the impetus to develop and deploy Smart Grid technologies, including wireless sensing, does not typically derive from cultural applications, the new connectivity and instrumentation of the built environment offer unique opportunities for culture, art and entertainment.

Unlike the Internet of Things, the Smart Grid's intrinsic focus on larger scale networks works well with the architectural design of places in the cultural sector—from public spaces to buildings and "third places". Employing wireless sensing

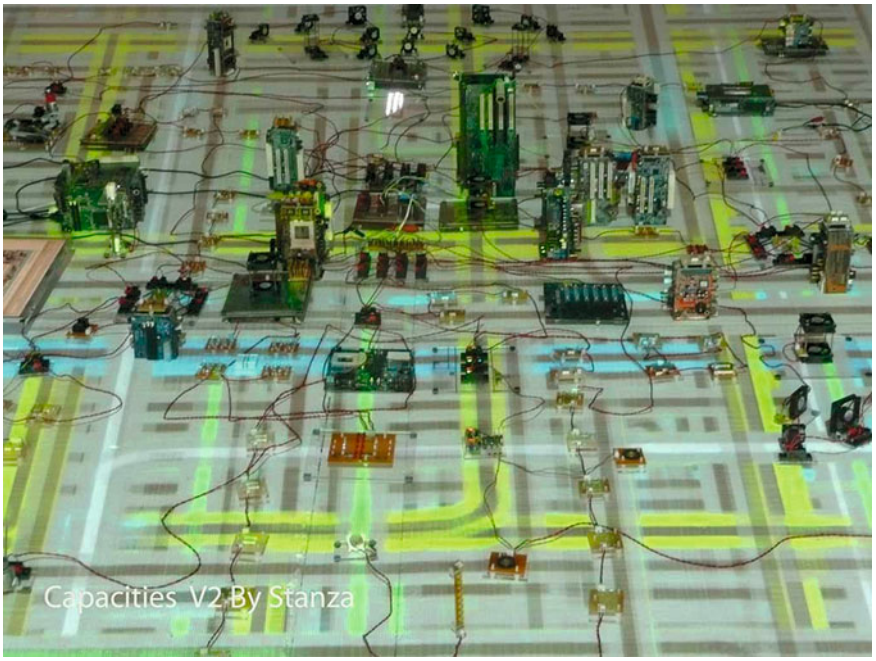


Fig. 3 Gallery photo of the artwork *capacities V2* by the artist stanza (photo by stanza)

within a Smart Grid approach could impact both the *design* of the built environment and its technological infrastructure, as well as generate new uses for an *already deployed* infrastructure. Such an infrastructure could be shared, duty cycling between supporting energy management and cultural applications, for example.

For many buildings, deployments of Smart Grid technologies are typically part of building projects or renovations. They often involve architectural design and physical construction; such projects typically consider each component and system within a design for its aesthetic impact and human usability, in addition to its functional role. In the past, the information technology of a building had often been exempt from aesthetic and (architectural) usability considerations, yet the close coupling of the physical and digital worlds in the Smart Grid suggests increasing similarity to other materials and systems in building projects, especially as cultural and artistic uses for these technologies emerge.

Rather than existing as fixtures attached to a finished structure, Smart Grid systems—including their wireless sensing components—will be considered integral to the fabric of buildings and contributors to the overall experience of the built environment. This is analogous to how lighting operates as not only a functional but fundamental aesthetic component of the experience of a place. The Internet-enabled sensing infrastructure of modern buildings will, over time, become a core component of their experience.

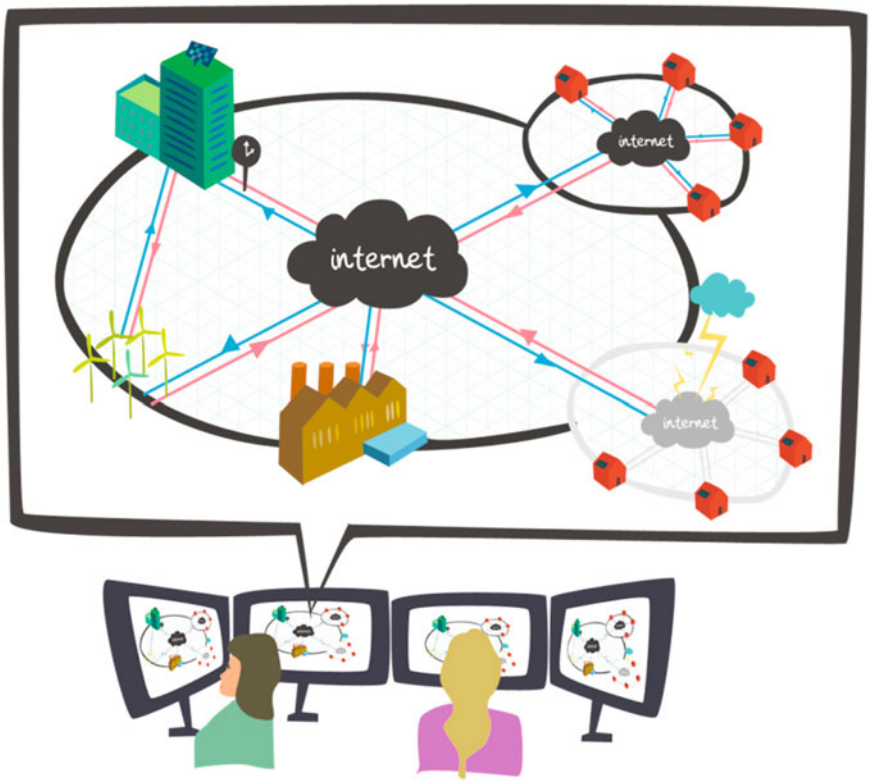


Fig. 4 Artist's rendering of the *smart grid*

More specifically, in Building Automation Systems (BAS), the emergences of Ethernet and IP as common networking platforms have already started to unify the mechanisms of control for aesthetic functions—such as window shades, lighting, and environmental sound—with building management. Modern buildings are likely to be equipped at construction with digitally controlled and often IP-addressable lighting and environmental systems (heating, ventilation, air conditioning), which is motivated at a minimum by energy management concerns. These systems can become part of the overall experience of the building as well. The same building may have security systems for intrusion detection and access control that employ a variety of presence, contact, flow, and identity sensing, the purposes of which may exceed management and extend to experiences with cultural or artistic objectives. At a rudimentary level, an apartment building's chilled water flow represents a measure of resource consumption that could be reflected in a display or other feature that engages the residents with their usage patterns.

In a more complex example scenario, wireless cameras providing personnel tracking could be employed to both monitor pedestrian flow and create an interactive experience. In fact, this has already been demonstrated using a wired solution;

Fig. 5 The Breezeway at Rockefeller Center by Electroland, in which LED lighting is controlled based on the movement of visitors, as sensed by a 3D camera-based personnel-tracking system. (photo by Electroland)

The Breezeway at New York's Rockefeller Center, created by Electroland using Tyzx's PersonTrack solution, implements an interactive lighting installation integral to the built environment. The Breezeway links visitor movement with aesthetic experience through a variety of real-time interactions and game-like relationships between movement and light. (See Fig. 5.) Another recent artwork, *Harmonic Fugue* is an interactive sound and light environment created by Christopher Janney. It employs a network of touch sensors in a hallway to trigger LED lighting and sound and encourage play in the environment.[5] (See Fig. 6.) While these projects currently rely on dedicated wired infrastructure, they might be implemented with a combination of wired and wirelessly networked devices and shared with other functions in the future.

The opportunities for multiple uses of sensing infrastructure promise a host of new challenges to future design of wireless sensing systems. Beyond the basic technological concerns, these challenges are similar to what can be observed in existing building systems. Specific authoring challenges (discussed further in the last portion of the chapter) include: application integration of IP-connected subsystems—the engineering challenge of providing applications access to various building subsystems across subnets, firewalls, and VLANs; privacy of sensor data gathered from individual participants; and the need for post-processing increasing amounts of sensor data to make it useful for applications. In addition to the ones discussed here, other security challenges abound in these increasingly heterogeneous networks that link

[5] http://janneysound.com/urban-musical-instruments/harmonic-fugue/

Fig. 6 *Harmonic Fugue* at Hendrix College, Conway, AK. © 2011, phenomenArts, inc.; Christopher Janney, artistic director. (photo: © N. Chenault)

critical and non-critical systems. These issues are not unique to cultural applications, but are foregrounded by that domain's explorations of linking typically disparate systems.

4 Public Space on the Grid

As illustrated above, public space provides a specific, important example of possible roles for wireless sensing, as part of a Smart Grid that supports cultural and artistic applications. They also yield a rich and important domain for experimentation, as they play an integral role in human life. As described by Oldenburg, public spaces provide another crucial type of *third place* between home and school or home and work, where people go for relaxation and exercise, social interaction, or solitude, for big and small events. Since Jane Jacobs' seminal book, *The Death and Life of Great American Cities* [16], challenged modern urban planning to understand neighborhood parks, value busy sidewalks, and develop mixed uses of urban space, the attention of planners, academics, and architects has been captured by the question of how to make great public spaces.

Designing public spaces that incorporate Smart Grid technologies offers an opportunity to engage with not only the basic functional requirements of the environment, but also the nature and potential of public space for cultural applications. In the same

Fig. 7 *The D-Tower* artwork (http://lab.v2.nl/projects/dtower.html) in the Netherlands, glowing green to reflect the current state of the surrounding people

way that the public broadcasting system was created to harness the technological innovations of television and radio for public benefit beyond what commercial markets were able to offer, a Smart Grid for cultural production could act as a public resource layered on top of emerging infrastructure. By supporting human–computer interaction as well as observation, wireless sensing can help to enliven public places with new learning, recreation, and interaction opportunities. Today, when people are in a public space, modern technology can easily connect them back to work or school, and enables commercial media to reach them there. Less attention has been given to the question of how technology can support the intrinsic and enjoyable roles of public space in lifelong learning, civic engagement, and social interaction. Below are reviews of examples that express sensor data in public space. As with those above, not all of these examples involve *wireless* sensing systems specifically. They illustrate opportunities that exist in the public sphere as the technologies become more viable for widespread deployment.

Cultural opportunities for public space include community-driven interactive art projects, such as the *D-Tower* in Doetinchem, The Netherlands, shown in Fig. 7. The *D-Tower* consists of an interactively-related physical sculpture, questionnaire, and web site that map the emotions of Doetinchem's inhabitants on a daily basis. The tower lights up in different colors in accordance with which emotions are most prevalent in the city, as measured by a daily survey of residents. This concept

can be extended to a broader range of possible data sources from a Smart Grid, suggesting an opportunity to engage the public with data about their environment in an aesthetically striking and ongoing manner. Other projects have engaged people in public space through interaction via SMS (text messages), including *CitySpeak* by Obx Laboratories at Concordia University,[6] which projected live text messaging in public space, and UCLA REMAP's *Junction/Juncture*[7] (2007), which used text messaged keywords from the public to control the selection of themes in a digital mural.

While the *D-Tower* engages participants to contribute to the state of a "beacon" in their public space, sensing can also be used to create direct relationships between the physical presence and movement of the public and artworks. Artist Rafael Lozano-Hemmer's "relational architectures" offer an example, using media to overlay individual, collective, or imagined memories onto physical architecture.[8] His recent artwork, *Nave Solar* (Fig. 8), uses sensing to capture the motion of participants and drive both a solar simulation and parametric sound environment. A very different project, *Color Forecast*[9] (Fig. 9), uses cameras in public space and basic computer vision techniques to "predict" fashion trends by observing the colors worn in Paris, Milan, and Antwerp. It links activity in public space to the web through sensing. Ross Miller's *Harbor Fog*[10] (Fig. 10) uses motion sensors to detect pedestrian interaction with an outdoor sculpture and generate changes in light and fog. Figure 11 shows *Great Street Games* by KMA,[11] which combined thermal imaging with projected light to create interactive urban playing areas.

These examples illustrate two important concepts. First, there are diverse opportunities for using sensing to enliven the cultural aspects of public space by creating new relationships among the public and their environment. Second, the intersection of sensing systems with media systems (as in the Internet of Things vision) will become increasingly important in public space applications specifically and for the Smart Grid vision more broadly.

Though most of the projects described deploy their own media infrastructure, there will be an ever-increasing set of opportunities to share existing infrastructure for cultural purposes. Increasingly, public spaces, like most buildings today, have broadband access to the public Internet as well as private IP networks. With the increasing importance of digital media, it is common for new facilities to have locally and remotely fed displays and projection, paging or sound systems, video recording or web broadcasting capabilities, and even large displays with touch or movement-based interfaces. These features represent opportunities for shared *output* infrastructure, in the same way that wireless and wired sensing can provide shared *input* infrastructure.

---

[6] http://www.obxlabs.net/

[7] http://la.remap.ucla.edu/junction/

[8] http://www.lozano-hemmer.com/

[9] http://www.pimkiecolorforecast.com/

[10] http://www.rossmiller.com/RM/harbor_fog.html

[11] http://www.kma.co.uk/

Fig. 8 Artist Rafael Lozano-Hemmer's *Nave Solar*, 2011. (Photo courtesy by Rafael Lozano-Hemmer)

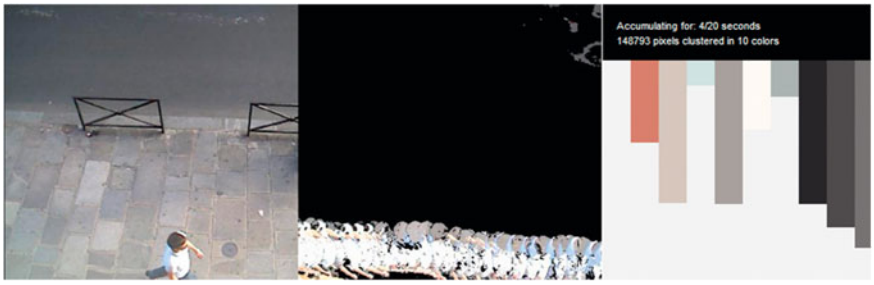


Fig. 9 The *Color Forecast* (http://www.pimkiecolorforecast.com/) project's view of a Paris street, showing camera signal, image after background subtraction, and color histogram

In projects with a "public face" (part of the building or facility near public streets or courtyards, or otherwise intended to engage visitors), such media systems provide unique opportunities and are often already available via local area TCP/IP networks. These media subsystems often support IP-based control and configuration, as well as media streaming over IP using common codecs (e.g., H264) or locally focused approaches, such as the CobraNet digital audio protocol [31]. Existing networking challenges strive to make these systems available for use in applications, a few of which are discussed in later sections of this chapter.

Fig. 10 *Harbor Fog* by Ross Miller, on Boston's Rose Kennedy Greenway

5 Participatory Sensing on Mobile Devices

The Smart Grid focuses on infrastructure integrated into the built environment. However, an even larger, more fluid "grid" already exists: the mobile phone network. Wireless sensing research has recently expanded to consider what is perhaps the most ubiquitous of wireless devices. While one of the most prominent non-communication roles of mobile devices in the commercial environment is as a *receiver* of content and applications, this research focuses on the typical smartphone's intrinsic sensing capabilities. Robust in construction and performance, widely deployed, and with an increasing number of sensors, mobile phones offer unique opportunities as they relate to scale and continual proximity to people. Various paradigms have emerged for the use of such devices to sense phenomena, including "opportunistic sensing" [6] and *participatory sensing*. The latter is used as the primary example here, because of its direct engagement with users and the public.

Participatory sensing tasks mobile devices to form interactive, participatory sensor networks. These networks enable public and professional users to gather, analyze and share local knowledge, as suggested in Fig. 12. They serve as paradigms for crowd-sourcing data collection to a population of volunteers, aiming to support both individualized feedback and/or compute aggregate models, maps, or statistics of mutual interest [5].

The ecosystem of sensing applications that use personal and community-scale participatory data collection continues to grow, fueled by the proliferation of mobile phones now accessible to large consumer populations. The concept of participatory

Fig. 11 Artist's rendering of KMA's *Great Street Games*

sensing has also expanded to incorporate smart residential wireless power meters (an overlap with the Smart Grid vision) in-vehicle GPS devices, sensor-enhanced entertainment platforms (e.g., Wii-fit), and activity-monitoring sportswear(e.g., the Nike+iPod system), which has reached mature (if niche) market penetration.

Research uses of participatory sensing that are transitioning into the mainstream include the collection GPS trajectories to monitor traffic patterns [14], collection of pollution traces to assess environmental impact [20], and gathering vehicular fuel-efficiency measurements to find "minimum footprint" driving routes [1]. While not addressed here in detail, vehicular networking will offer another new front for cultural applications. Ubiquitous wireless connectivity (e.g., WiMax) and vehicular Internet access enable new applications that exploit networking to export, aggregate, and exploit sensory information.

Many culturally-engaged participatory sensing examples are in an area termed by Paulos et al. [23] as "participatory urbanism," in which mobile devices act as networked measurement instruments used to understand the urban environment. These authors provide examples of observational uses of wireless sensing devices, e.g., extending cell phones to act as air quality sensors. Such uses are able to have a complementary intent to the artworks described above, another significant intersection of wireless sensing with public space: Wireless sensing is deployed to explore our surroundings rather than as part of an intervention. There is also an important crossover with the creation of original artworks in this public context. For example,

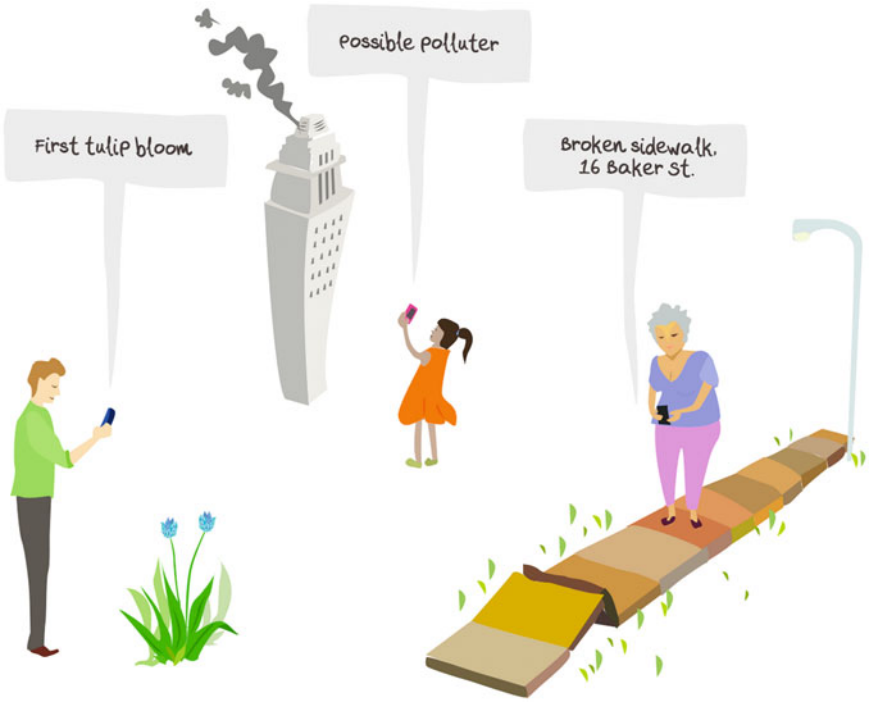


Fig. 12 Artist's rendering of the concept of *participatory sensing* on mobile phones

Preemptive Media, a group of "artists, activists, and technologists" (Beatriz da Costa, Jamie Schulte, and Brooke Singer) created the project *Area's Immediate Reading* (*AIR*), in which portable air monitoring devices with CO, NOx, and O_3 sensors are used to explore neighborhoods for pollution hotspots.[12] These devices and their use on the street are shown in Fig. 13. Indeed, many "participatory" projects involve some type of "output" back into the urban environment.

AIR is representative of an important trend for not only participatory sensing, but also for the other perspectives on wireless sensing described above. As sensing systems become more ubiquitous, they will be used for applications that cross between many spheres, e.g., productivity, sustainability, learning, and culture. They will also be tools for engagement—the citizenry to their surroundings from scientific, social, civic, and cultural perspectives. Paulos et al. [24] discuss what they call the "rise of the expert amateur," in which non-technologists leverage technologies, such as wireless sensing, to conduct investigations of their environment that are not easily categorized within existing frameworks.

[12] http://www.pm-air.net

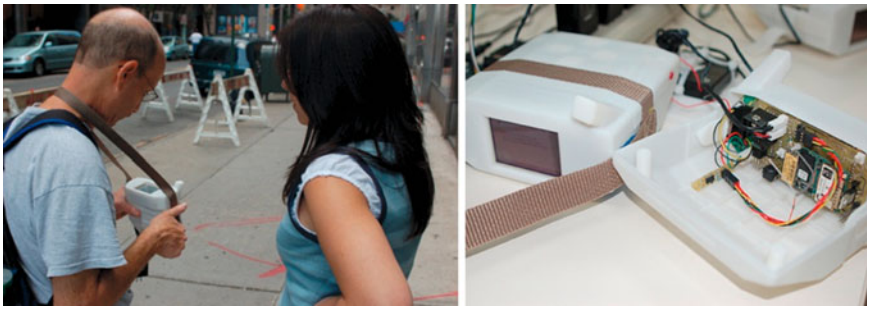


Fig. 13 Portable air monitoring device: the preemptive media project's *AIR* (*Area's Immediate Reading*). (Photos by preemptive media.)

6 Challenge of Personal Privacy

Perhaps one of the most pressing challenges for participatory sensing on mobile devices is privacy protection for individual participants. For example, a significant amount of work has focused on the capture and analysis of individual mobility patterns (time-location "traces") as fundamental data streams around which to organize and correlate other collected data [28]. Detailed time-location traces are obtained easily because modern mobile phones are often GPS-equipped and commonly carried by owners almost continuously. Such traces are of substantial and immediate interest for studying time-location patterns of individuals and communities in urban planning, public health, social and behavioral sciences, and cultural domains.

Time-location traces provide a rich data set that can be used and can also reveal significant details of a person's life. A complicated challenge is presented by balancing ease of use, user participation in privacy decisions, and creator's and scientist's desires for detailed data sets of this and similar types. This and other participatory sensing privacy challenges are surveyed [7] and [27], and discussed in more detail below, as are challenges in networking of mobile data publishers and applying machine learning to participatory sensing data sets.

6.1 Event-Based Perspectives: Intersections with Live Performance

The three perspectives above focus on *objects, infrastructure and buildings,* and *mobile devices*, respectively. Cultural applications often take an *event-based* perspective as well, whether within an ad-hoc event, such as a "flash mob", or a more formal one, such as a live performance (illustrated in Fig. 14). Here, the latter serves as a representative example of the event-based perspective, providing unique opportunities for developing new modes of human-computer interaction.

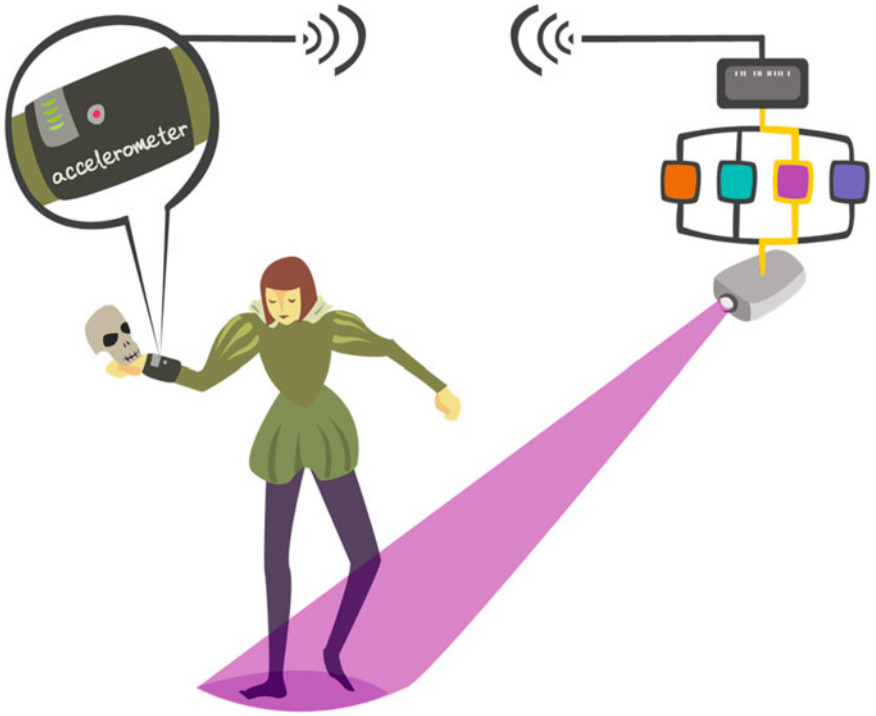


Fig. 14 Artist's rendering of live performance using wireless sensing

During the last 5 years, reliable wireless sensing has become increasingly accessible to creators of live performance through some of the same do-it-yourself (DIY) means discussed above—include motes, easily programmed microcontrollers, wireless radio modules, and other inexpensive, easily integrated components. Previously, use of wireless sensing had been limited to experimental performances that could tolerate less reliability (or scaling) or large budget productions that could afford extensive development and testing.

Live performance provides a context for wireless sensing that is already infused with a wide range of digital technology at a variety of scales. Theater and dance artists, as well as musicians, already incorporate digital media into their work; they use digital systems for playback, control and manipulation of sound, lighting, motor, and projection effects. In fact, creators of performance have experimented with sensing itself in several contexts. In the eminent choreographer Merce Cunningham's piece *Biped* (1999), he used motion capture quite differently than in Robert Zemeckis' more familiar movie *Polar Express* (2004), which used motion capture for facial and body animation. To striking effect, Cunningham juxtaposed live dancers with the projection of non-photorealistic renderings generated from captured performance data. [8, 19] Work like his has expanded interest within the performance world in marker-less tracking and other long-term computer vision research, which will be

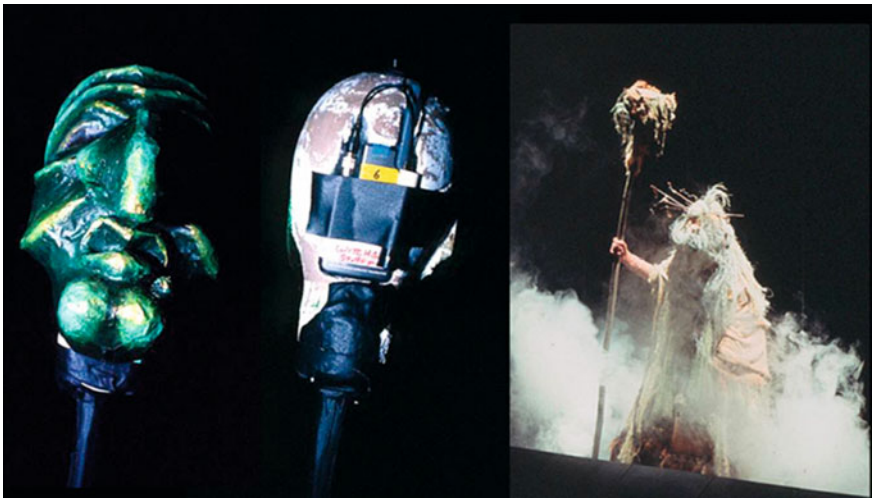


Fig. 15 Wireless position sensor embedded in prop, shown onstage in the UCLA Department of Theater's 2001 production of Ionesco's *Macbett*. (Courtesy of UCLA REMAP)

supported by pervasive computing infrastructures, and have already been explored by Sparacino [29] and others. As wireless sensing becomes viable, these vanguard works provide inspiration for its use in performance, perhaps enabling what had to be pre-rendered for Cunningham to be created in real-time.

An established, though still experimental, real-time use of wireless sensing systems in live performance is to map relevant physical phenomena—movement, sound, and image from the performers and sometimes the audience—into some of these already digitally controllable outputs of the stage environment, including lighting, sound, and projection. In this use case, the phenomena are amplified or transformed in real-time to create an effect under direct control of the performer or audience, without requiring an operator. For example, the author's work in a UCLA theatrical production of Ionesco's *Macbett* [3] is shown in Fig. 15. There, a commercial ultrasonic positioning system was deployed to track the position of actors and props, such as the witch's staff shown in the figure. The resulting position data was processed to yield short-time features that were then used to put lighting and sound under direct control of the former. Work by many artists—such as David Rokeby and the Troika Ranch and Palindrome dance companies—has explored similar uses of such technology. More recently, Park et al. [22] created a wearable wireless sensor platform for heartbeat, joint angle, gyro, and accelerometer to support interactive dance performances. Feldmeier and Paradiso [9] created an inexpensive ("give-away") wireless sensor suitable for gathering rhythm and activity data from crows or large ensembles. Commercial companies, such as beaudry interactive, have started to integrate high-performance, inertial wireless sensing into theatrical props, as shown in Fig. 16.

Fig. 16 Wireless gyro- and accelerometer-enabled wands for theatrical performance. (Courtesy of beaudry interactive)

7 Challenges

Direct, real-time mappings between the data generated by these wireless sensing systems and aspects of the stage environment, such as lighting and sound, are common. They can be done with minimal software development, in contrast with gesture recognition or other approaches requiring machine learning or other advanced techniques. However, the aesthetic limitations of one-to-one real-time mappings are significant. As discussed below, machine learning is an important area of future development. It could enable more sophisticated mappings from the action of performers and audience to stage control.

Additionally, unlike the intermittent, ongoing operation of mobile sensing and the continuous but lower sampling rate sensing of many Smart Grid applications, performances most often consist of an intense few-hour period of continuous operation with high reliability and sampling rate requirements. (After this period, devices can be checked and batteries replaced before another performance or rehearsal.) In this way, they have the most in common, perhaps, with the requirements of commercial gaming and entertainment that incorporate sensing, such as the Nintendo Wii Remote and Microsoft Kinect. While device performance is not treated explicitly in this

chapter, the use of wireless sensing in live events provides a challenging set of requirements for future research, especially when used as a basis for human-computer interaction with professional performers [4].

Live performances are also created within a temporal structure shared with other temporary deployments and iteratively developed projects: cycles of design, rehearsal / testing, and public demonstration. One of the most important periods to support is rehearsal, in which various options for performance and design are attempted, revised and refined in a continuous cycle of iterative development with many other elements. Based on the author's experience, rehearsal presents an extremely demanding use case for wireless sensing systems, which in this context must include authoring interfaces, middleware, and physical packaging in addition to basic electronic components. While some reliability requirements are relaxed in a rehearsal scenario, the ability to reconfigure systems while they are online is of paramount importance—and not always achievable using current technologies. Unlike performance itself—in which some elements are typically fixed, even in the most improvisational circumstances—rehearsals are a "learn-by-doing" environment, in which a constantly shifting set of ideas are explored. The more "viscosity" to change presented by the technological systems of performance, the more limited their integration into the artwork becomes. As a result, performance can provide a unique driver towards flexibly configured and controlled systems, as well as suggest new, iterative, and online authoring approaches to human-computer interaction.

7.1 Research Opportunities: Authoring Cultural Applications that Use Wireless Sensing

As is evident in the case of live performance, and also true for the other perspectives described above, a fundamental challenge in incorporating wireless sensing into cultural applications is the need for *authoring frameworks* that provide appropriate abstractions and flexible approaches to on-the-fly configuration and experimentation.

For digital video to move into the mainstream of cultural production during the past 15 years, not just hardware and software for capture and dissemination were required. Editing software becoming available and relatively easy to use was essential. To be most useful in cultural production, wireless sensing, viewed from any of the perspectives outlined, also requires tools and techniques for creative authoring that go beyond simply making the sensors network-addressable, power-efficient and reliable.

From the wide variety of examples given above, it is clear that sensing systems can be integrated into diverse cultural projects, created with broad considerations that range from aesthetic, technical, and usability-related concerns to specific challenges from deployment contexts, such as urban public space. Given the holistic, iterative way in which many such cultural projects are designed, components and systems must not only be conceived for their final role in a public experience, but for their use by designers and creators. Through iterative development, design decisions are tested against a variety of factors.

In this way, authoring and deployment in a cultural context is a "tussle" between three types of goals.

Operational objectives relate to familiar requirements for performance and robustness (amongst others).

Expressive or communicative goals emerge from applications' important, non-functional considerations: lighting color quality or cross-fade smoothness when driven from a sensor input; use of multiple modalities of media; the size and thickness of a device; or the openness of a platform to user configuration. These goals are connected not by basic function but by the needs of iterative development, end-user experience and deployment context. As described above, cultural applications place high demands on integration across heterogeneous systems. This, for example, suggests that high-bitrate content and sensing will not be so separate in the future.

Participatory goals concern the demands of user-facing components (for example, when users may be the designers of cultural applications, not just an audience), just-in-time configuration and content selection, ad-hoc networking, and consideration of intermittent interaction and other human factors.

Stated elsewhere are a few general design considerations for cultural applications that are valuable for wireless sensing systems [5]. They can be applied across operational, expressive, and participatory goals:

- A "sense of time," the consideration of synchronization of events (an operational perspective) or their rhythm (an expressive perspective);
- A "sense of space" that extends beyond location awareness to the notion of a defined region of responsibility for a given sensor or system;
- "Drill-down," the opportunity for abstraction layers to be set aside by designers who wish to deal with less processed data directly (an expressive objective sometimes at odds with operational benefits of increased abstraction);
- "Guided deployment," in which systems aid the users directly in deploying distributed sensor networks to meet appropriate coverage and other goals.

Each of these considerations can be applied to the various perspectives described above. For example, a "sense of space" in Smart Grid sensing infrastructure could come from providing location metadata (both building name and GPS coordinate) for each sensor plus spatial query capability as a straightforward application-programming interface. In this manner, a cultural application might easily address all sensors within a given location through a building name query rather than referring to an external, offline mapping between sensor IP address, port, and/or channel number. While such mappings are straightforward conceptually and exist in research labs and certain larger systems, they are often not yet available in practice, and often limit the ability to quickly author applications with a human-understandable notion of the location of wireless sensor data.

8 Evaluating Authoring Systems

To evaluate these and other research results—intended to reduce the complexity of authoring—systematic methods can be applied that are inspired by other domains. An example is the cognitive dimensions framework, originally developed for evaluating programming languages but applied to other domains [12, 13]. Six framework dimensions of note that apply to authoring cultural applications of wireless sensing systems are described below:

- *Closeness of mapping* between the "problem world" (mental models) and the "real world" in the systems to be designed and deployed (close mappings reduce authoring complexity but can come at the cost of generality).
- *Secondary notation*: Expressiveness beyond the official semantics. (This provides an opportunity to embed useful information consistently, in code and configuration, for example.)
- *Hard mental operations:* How much must be expressed / manipulated outside of the notation. (Typically this should be reduced, balanced for the possible further losses of generality).
- *Premature commitment*: Constraints on the order of doing things (especially important to limit in iterative development and testing typical to cultural applications).
- *Progressive evaluation*: Ability to work with incomplete systems (similarly vital for cultural applications).
- *Viscosity*: Resistance to (local) change. (For many types of systems that have multiple uses and evolving configurations, this should be reduced).

Outlined below are three important research areas for promoting the cultural use of wireless sensing systems with discussion based on these dimensions. Examples of how "lower-level" (non-application) research can benefit cultural applications of wireless sensing are provided.

9 Machine Learning

Though not specific to wireless sensing, machine learning will play a crucial role in increasing the use of wireless sensing in cultural applications. The primary producers of cultural experiences are not scientists, engineers, or statisticians. They generally do not have experience in deriving data they want from sensed signals using machine learning and signal processing techniques. Toolkits for machine learning will help enable them to do so, improving the closeness of mapping between mental models (of gesture, for example) and programming interfaces (to body-mounted accelerometers, for example).

Another example is the FAAST toolkit,[13] developed by the USC Institute for Creative Technologies. It provides basic gesture recognition from the Microsoft Kinect

[13] http://projects.ict.usc.edu/mxr/faast/

3D camera and maps recognized gestures to keystrokes and mouse movements. This enables artists and others to easily experiment with computer vision-based interaction. Toolkits like FAAST already benefit human-computer interaction uses of wireless sensing for cultural applications by simplifying the amount of programming necessary to map gestures to other parameters. By enabling "learning by example", the *closeness of mapping* between the problem domain and the control domain is increased, though the training time for the system could increase *premature commitment* and negatively impact both expressive and operational objectives.

In wireless sensing for *observation*, learning of high-level patterns can be used to develop measures of difference between repeated actions (e.g., an actor's movement from rehearsal to rehearsal or performance to performance) or to extract higher-level features that are more relevant to the application at hand (e.g., significant places in a person's daily commute captured as part of a participatory sensing activity. [26]) Additionally, managing the reputation of participants in systems—relying on contributions of varying quality from many individuals—can also leverage learning of contribution patterns [25].

10 Networking

Similarly, IP has provided consistent and cost-effective connectivity to building systems, traditional hosts, and mobile devices, but connectivity alone does not make authoring accessible for the various cultural applications outlined. The Internet of Things and Smart Grid visions chart a course for increasing crossover between Internet applications and embedded computing, sensing, and control. However, just because systems are increasingly IP-connected does not mean that it is easy to create and deploy secure applications that integrate their various components, especially with the commodity Internet. Device networking advances can be applied to improve the *closeness of mapping*, make *secondary notation* more relevant to the application at hand, and reduce *premature commitment* due to *viscosity* of networking configuration changes.

In practice, physical segmentation and/or address translation in IP networks (used in the applications discussed above) often make their configuration difficult to author with and brittle to application change, especially in contexts without significant network engineering support. Or, following Green's terminology, it requires a number of *hard mental operations* to track the relationship between, for example, IP network configuration and application needs. For reasons related to security, stability, QoS, and simplicity of administration, the many IP networks involved in both infrastructure-based and mobile sensing are often isolated through VLANs, with firewalls between systems, or are not interconnected at all. Networks for building automation and media distribution (to form crucial components of future cultural applications, as described above) are often similarly segregated. They use different protocols to share even basic data and control mechanisms, and can be quite hard to integrate at the scale of an enterprise or in specialized facilities. Differing

protocols and IP gateway mechanisms, addressing schemes, content distribution and security requirements across various subsystems pose further authoring challenges to software developers.

Furthermore, in current systems, significant knowledge is bound up in areas not accessible to the application software developer or end user. VLANs embody boundaries between systems identified during design and deployment, but they are typically unseen or inaccessible, as are IP sub-netting and routing, which reflect device organization and interconnectivity. Firewall configurations describe expectations for access between systems. Keys and certificates for SSL connections and VPNs may identify components or connections. VPN configuration and enterprise authentication hold network access permissions. None of these are typically accessible to application software in traditional systems; they are "network configuration". In fact, they represent important system information that is often replicated ad-hoc from configuration to configuration. A simple example is how an application must be configured to know that one IP subnet is for lighting control and another is for network streaming of sound, and must be connected physically to both. This is site-specific and meaningless to the application, and might not be generalized or abstracted in a quickly assembled system.

Information-Centric (or *Content-Centric*) Networking is an example of recent research that could be applied to reduce the complexity of authoring applications for distributed systems of wireless sensors. One such project in this area is the *Named Data Networking* (NDN) effort led by UCLA and Xerox PARC. NDN is a communication architecture based on named content that aims to generate a next-generation, evolutionary replacement for TCP/IP [21]. Rather than addressing content by its location, NDN refers to it by name. The project's approach is based on Content-Centric Networking research at PARC [17], and similar to other CCN and ICN projects surveyed [11].

In NDN, the network infrastructure routes and forwards based directly on hierarchical names that can be selected to have meaning to the application, e.g., <enterprise_root>/occupancy/<building>/<room> for a networked occupancy sensor for a given building and room. Though similar results can be achieved using DNS, no latency is added due to lookup, and integration with intranet DNS management is not necessary. Additionally, while NDN supports TCP/IP and UDP/IP as underlying transport protocols, it is not reliant on them. It could also, for example, take advantage of broadcast capability in modern wireless channels to discover devices and communicate without requiring address assignment. By improving the closeness of mapping between the network and application semantics through the use of names, it enables sensors and devices to be addressed by both the network and the application author in terms of application-understandable location, function, or other identifiers, without the need for middleware. While still an open topic of study, in this way, NDN is an example of how lower-level research in fundamental areas, such as networking, can aid authors of cultural systems.

11 Privacy

Finally, many cultural applications involve direct interaction between or observation of participants and the public. Especially in mobile phone-based participatory sensing—wherein sensing "follows" people anywhere that they take their mobile device—there are significant privacy challenges, both technical and social. The proximity to people's everyday activities, the subtlety with which sensing can be accomplished, and the rapid dissemination of sensed data together create significant new avenues for unexpected and/or uncontrolled leakage of personal information. A full review of privacy for each perspective described above is provided outside of the scope of this chapter, but cultural applications provide a unique venue for privacy research that would benefit from further investigation. Based on the author's experience, cultural applications are likely to be created in contexts without formal privacy and security policies. This leaves, for example, limits on data retention or de-identification to ad-hoc decision-making processes, if they are dealt with explicitly at all. In this case, their *operational objectives* take precedence over *participatory goals*.

Formalizing privacy and security requirements for sensing applications is a matter of both policy and technology, as discussed by Shilton [27] in the context of participatory sensing and Kang et al. [18], relative to self-surveillance using similar technologies. Shilton provides a framework for system design that includes three factors—*participant primacy*, *longitudinal engagement* and *data legibility*—intended to engage users in data-sharing decisions and protect their privacy in a participatory sensing context. While not all cultural applications are explicitly participatory, their (typical) engagement with lay public audiences suggests that these dimensions can provide a provocative starting point for further research. "Participant primacy" is to foreground the role of the sensed participant in deciding what is done with their data. "Longitudinal engagement" suggests that those being sensed should be involved throughout the preparation, collection, and retention of data about them. "Data legibility" emphasizes the importance of making collected data understandable to sensed participants. It also promotes important cognitive benefits for authors, including increased *closeness of mapping* and a reduction in *hard mental operations* to understand system function. These dimensions are designed to give participants in sensing systems a substantial decision-making role with respect to their data.

Given that the object (Internet of Things), infrastructure (Smart Grid) and mobile (participatory sensing) perspectives on wireless sensing all promote ongoing interaction between people and the network through sensors, these design dimensions developed for continuous participatory sensing can be adapted to the broader context of this chapter. For example, participant primacy suggests that a participant is the owner of a smart, Internet-connected object who should have access to the data that it gathers about them. This should be in contrast to the data being available solely to the entity that manufactured the object, wishing to integrate it with their social networking presence. Developing mechanisms for longitudinal engagement with data collection in smart infrastructure might motivate the building of a deeper

engagement with users about key issues, e.g., how their environment "sees" and "remembers" them. To each perspective, legibility applies a goal of making what is sensed available in a clear and understandable way to the end user—an approach that can yield a deeper grasp for developers and system designers as well.

Incorporation of privacy-related design goals may be motivated by the individual beliefs of creators, corporate social responsibility or civic responsibility. A challenge to wireless sensing researchers exists not only in the mechanics of privacy-preserving approaches—summarized in recent work described above—but in how they are presented to the end users and, even before that, to developers. Similar to machine learning and networking, providing effective, participatory options for privacy are an authoring challenge. Further research in the field could aim to provide tools that enable key privacy questions, to be addressed effectively during design time.

12 Conclusion

Wireless sensing—whether incorporated in objects, infrastructure or mobile devices, or provisioned across each of these on an event-by-event basis—stands to become a significant element of cultural expression. It enables new types of human-computer interaction, as well as automated observation of built and natural systems that can be incorporated into cultural production.

Creators of the range of applications, surveyed by this research, face challenges that, while relevant to other domains, are not often considered primary concerns. In many ways, however, they anticipate the challenges of broader uses of wireless sensing. Other domains stand to benefit significantly from the results of research driven by the integrative, iterative approaches taken to design and develop cultural projects. Support of authoring, in particular, represents an area where there are advances in learning, networking, and privacy. These can be applied to benefit creators of cultural experiences, as well as other users of wireless sensing systems.

Acknowledgments The author wishes to thank UCLA REMAP and, in particular, Ren Rong for her illustrations and Weihan Zhou, Zening Qu, Tina Xie, and Xiyuan Xiao for their contributions to this chapter.

References

1. T. Abdelzaher, *Green GPS-assisted Vehicular Navigation* (Handbook of Energy-Aware and Green Computing Chapman & Hall/CRC, 2012)
2. M.S. Amin, B.F. Wollenberg, Toward a smart grid: power delivery for the 21st century. IEEE Power and Energy Magazine **3**(5), 34–41 (2005)
3. J. Burke, *(2002) Dynamic performance spaces for theatre production* (Theatre Design & Technology, Winter, 2002)

4. J, Burke, D. Estrin, M. Hansen, A. Parker, N. Ramanathan, S. Reddy, M. B. Srivastava, Participatory sensing. World Sensor Web Workshop, ACM Sensys Boulder, Colorado, 31 Oct 2006
5. J. Burke, J. Friedman, E. Mendelowitz, H. Park, M.B. Srivastava, Embedding expression: Pervasive computing architecture for art and entertainment. J. Pervasive and Mobile Computing **2**(1), 1–36 (2006)
6. Campbell, A. T., Eisenman, S.B., Lane, N.D., Miluzzo, E., & Peterson, R.A. (2006) People-centric urban sensing. Proc. 2nd Intl. Workshop on Wireless internet (WICON '06).
7. D. Christin, A. Reinhardt, S.S. Kanhere, M. Hollick, A survey on privacy in mobile participatory sensing applications. J. Systems and Software **84**(11), 1928–1946 (2011)
8. A. Dils, The ghost in the machine: Merce Cunningham and Bill T. Jones, Performing Arts Journal **70**(2002), 94–104 (2002)
9. M. Feldmeier, J. Paradiso, An Interactive Music Environment for Large Groups with Giveaway Wireless Motion Sensors. Computer Music Journal **31**(1), 50–67 (March 2007)
10. N. Gershenfeld, R. Krikorian, D. Cohen, The Internet of Things. Scientific American **291**(4), 76–81 (2004)
11. Ghodsi, A., Shenker, S., Koponen, T., Singla, A., Raghavan, B., & Wilcox, J. (2011) Information-centric networking: seeing the forest for the trees. Proc. 10th ACM Workshop on Hot Topics in, Networks (HotNets-X).
12. Green, T. R. G. (2000) Instructions and descriptions: some cognitive aspects of programming and similar activities. Invited paper, in Di Gesù, V., Levialdi, S. and Tarantino, L., (Eds.) Proc. Working Conference on Advanced Visual Interfaces (AVI 2000). New York: ACM Press, pp 21–28.
13. T.R.G. Green, M. Petre, Usability Analysis of Visual Programming Environments: A 'Cognitive Dimensions' Framework. Journal of Visual Languages and Computing. **7**(2), 131–174 (1996)
14. Hull, B., Bychkovsky, V., Zhang, Y., Chen, K., Goraczko, M., Miu, A., Shih, E., Balakrishnan, H., & Madden, S. (2006) CarTel: a distributed mobile sensor computing system. Proc. 4th Intl. Conf. on Embedded networked sensor systems (SenSys '06). ACM, New York, NY, USA, 125–138.
15. Ishii, Hiroshi. (2008) "Tangible Bits: Beyond Pixels". Proc. 2nd International Conference on Tangible and Embedded Interaction (TEI'08), Feb 18–20 2008, Bonn, Germany.
16. J. Jacobs, *The Death and Life of Great American Cities* (Vintage, New York, 1961)
17. Jacobson, V., Smetters, D., Thornton, J., Plass, M., Briggs, N., & Braynard, R. (2009) Networking named content. ACM CoNEXT, 2009.
18. J. Kang, K. Shilton, D. Estrin, J. Burke, M. Hansen, Self-Surveillance Privacy. Iowa Law Review **97**, 809 (2012)
19. Kisselgoff, A. (1999) Lincoln center festival review: A youth of 80 kicks up his heels, in: New York Times Section E, Part 1, July 23, 1999.
20. Mun, M. Y., Estrin, D., Burke, J. & Hansen, M. (2008) Parsimonious Mobility Classification using GSM and WiFi Traces. HotEmNets 2008.
21. Named data networking project (NDN). (2012) http://named-data.org. Retrieved Apr. 2012
22. Park, C., Chou P. H., & Sun, Y. (2006) A Wearable Wireless Sensor Platform for Interactive Art Performance. Proc. Fourth Annual IEEE International Conference on Pervasive Computing and Communications (PerCom, March 13–17, 2006 (Pisa, Italy, 2006), pp. 52–59.
23. Paulos, E., Honicky, R.J., & Hooker, B. (2008) Citizen Science: Enabling Participatory Urbanism. Handbook of Research on Urban Informatics: The Practice and Promise of the Real-Time City. Foth, M., ed. Hershey, PA: Information Science Reference, IGI Global, 2008.
24. Paulos, E., Kim, S., & Kuznetsov, S. (2001) The Rise of the Expert Amateur: Citizen Science and Micro-Volunteerism. Social Butterfly to Engaged Citizen: Urban Informatics, Social Media, Ubiquitous Computing, and Mobile Technology to Support Citizen Engagement. Edited by: Foth, M., Forlano, L., Satchell, C., & Gibbs, M. Cambridge, MA: MIT Press. 2011.
25. Reddy S., Estrin, D., & Srivastava, M. (2010) Recruitment Framework for Participatory Sensing Data Collections. Proc. Intl. Conf. on Pervasive Computing (Pervasive), May 2010.

26. S. Reddy, M. Mun, J. Burke, D. Estrin, M. Hansen, M.B. Srivastava, Using Mobile Phones to Determine Transportation Modes. ACM Transactions on Sensor Networks **6**(2), 1–27 (2010)
27. K. Shilton, Four billion little brothers?: privacy, mobile phones, and ubiquitous data collection. Commun. ACM **52**(11), 48–53 (2009)
28. Shilton, K., Burke, J., Estrin, D., Hansen, M., Kang, J., & Govindan, R. (2009) Designing the Personal Data Stream: Enabling Participatory Privacy in Mobile Personal Sensing. 37th Research Conference on Communication, Information, and Internet Policy (Telecommunications Policy Research Conference). September 25–27, 2009. Arlington, Virginia.
29. F. Sparacino, C. Wren, G. Davenport, A. Pentland, (1999) Augmented performance in dance and theater. International Dance and Technology, *Tempe* (Arizona, February, 1999). 1999
30. M. Weiser, *(1991) The Computer for the 21st Century* (Scientific American, September, 1991)
31. Yamaha Commercial Audio. (2007) An introduction to networked audio. http://www.cobranet.info/sites/default/files/YamahaWP_-_Intro_to_networked

Editor Biography

Habib M. Ammari is an Associate Professor and the Founding Director of Wireless Sensor and Mobile Ad-hoc Networks (WiSeMAN) Research Lab, in the Department of Computer and Information Science, College of Engineering and Computer Science, University of Michigan-Dearborn, since September 2011. He obtained his second Ph.D. degree in Computer Science and Engineering from the University of Texas at Arlington, in May 2008, and his first Ph.D. in Computer Science from the Faculty of Sciences of Tunis, in December 1996. He has a strong publication record in top-quality journals, such as ACM TOSN, ACM TAAS, IEEE TPDS, IEEE TC, Elsevier COMNET, Elsevier PMC, Elsevier JPDC, Elsevier COMCOM, and high-quality conferences, such as IEEE SECON, IEEE ICDCS, EWSN, and IEEE MASS. He published his first Springer book, "Challenges and Opportunities of Connected k-Covered Wireless Sensor Networks: From Sensor Deployment to Data Gathering" in August 2009. Also, he is the author and editor of two Springer books, "The Art of Wireless Sensor Networks: Fundamentals" and "The Art of Wireless Sensor Networks: Advanced Topics and Applications", which will be published in 2014. He has been selected for inclusion in the AcademicKeys Who's Who in Engineering Higher Education in 2012, the AcademicKeys Who's Who in Sciences Higher Education in 2011, Feature Alumnus in the University of Texas at Arlington CSE Department's Newsletter in Spring 2011, Who's Who in America in 2010, and the 2008-2009 Honors Edition of Madison Who's Who Among Executives and Professionals. He received several prestigious awards, including the Certificate of Appreciation Award at ACM MiSeNet 2013, the Certificate of Appreciation Award at the IEEE DCoSS 2013, the Certificate of Appreciation Award at the ACM MobiCom 2011, the Outstanding Leadership Award at the IEEE ICCCN 2011, the Best Symposium Award at the IEEE IWCMC 2011,

H. M. Ammari (ed.), *The Art of Wireless Sensor Networks*,
Signals and Communication Technology, DOI: 10.1007/978-3-642-40066-7,

the Lawrence A. Stessin Prize for Outstanding Scholarly Publication from Hofstra University in May 2010, the Faculty Research and Development Grant Award from Hofstra College of Liberal Arts and Sciences in May 2009, the Best Paper Award at EWSN in 2008, the Best Paper Award at the IEEE PerCom 2008 Google Ph.D. Forum, the Best Graduate Student Paper Award (Nokia Budding Wireless Innovators Awards First Prize) in May 2004, the Best Graduate Student Presentation Award (Ericsson Award First Prize) in February 2004, and Laureate in Physics and Chemistry for academic years 1987 and 1988. Also, he was selected as the ACM Student Research Competition Finalist at the ACM MobiCom 2005. He is the recipient of the Nortel Outstanding CSE Doctoral Dissertation Award in February 2009, and the John Steven Schuchman Award for 2006-2007 Outstanding Research by a PhD Student in February 2008. He received a three-year US National Science Foundation (NSF) Research Grant Award, in June 2009, and the US NSF CAREER Award, in January 2011. He is the Founding Coordinator of both of the Research Colloquium Series since September 2011, and the Distinguished Lecture Series since January 2012, in the College of Engineering and Computer Science at the University of Michigan-Dearborn. He has been invited to give invited talks at several reputed universities. He is the Founder of the ACM Annual International Workshop on Mission-Oriented Wireless Sensor Networking (ACM MiSeNet), which has been co-located with ACM MobiCom since 2012. He serves as Associate Editor of several prestigious journals, such as ACM TOSN, IEEE TC, and Elsevier PMC. Also, he has served as Program Chair, Session Chair, Publicity Chair, Web Chair, and Technical Program Committee member of numerous ACM and IEEE conferences, symposia, and workshops.