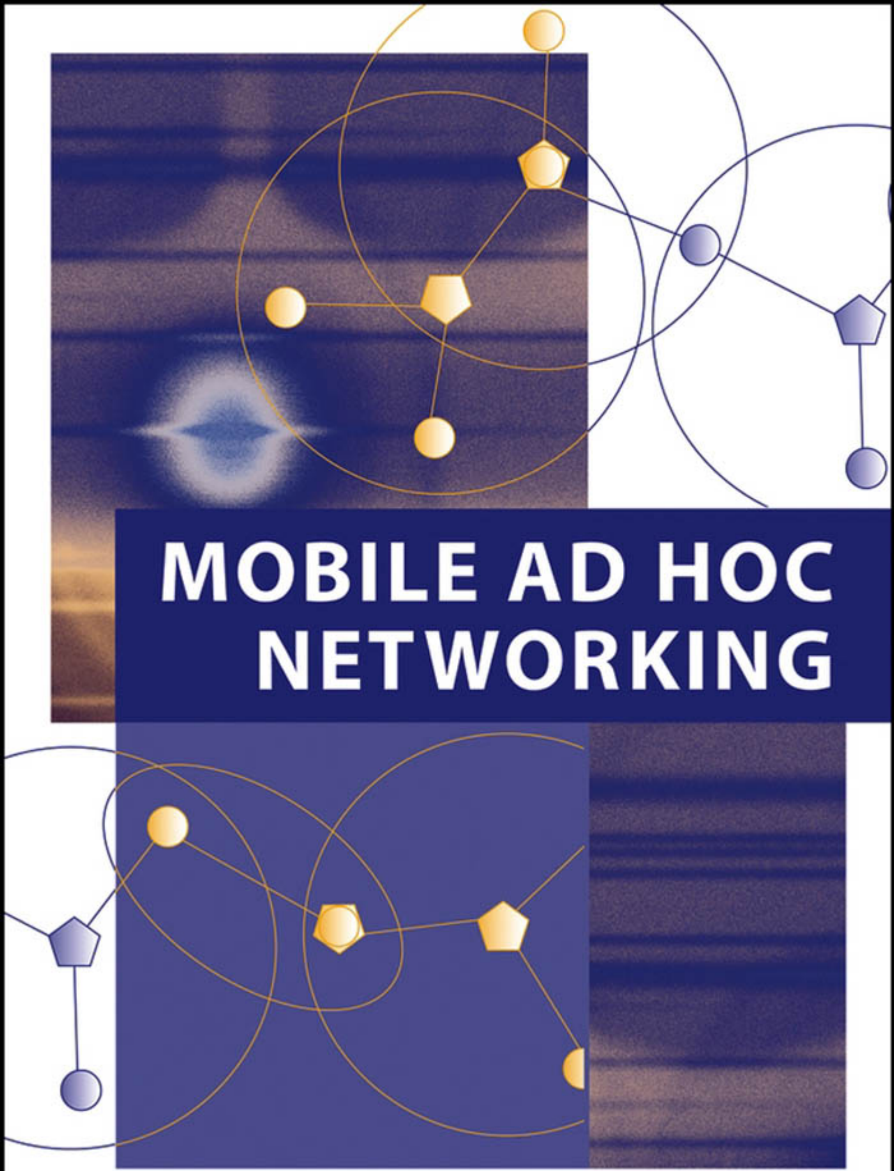


MOBILE AD HOC NETWORKING

STEFANO BASAGNI • MARCO CONTI

SILVIA GIORDANO • IVAN STOJMENOVIĆ

MOBILE AD HOC NETWORKING

Edited by

STEFANO BASAGNI
Northeastern University

MARCO CONTI
Italian National Research Council (CNR)

SILVIA GIORDANO
University of Applied Science, Switzerland

IVAN STOJMENOVIC
University of Ottawa

IEEE

IEEE PRESS

WILEY-
INTERSCIENCE

A JOHN WILEY & SONS, INC., PUBLICATION

MOBILE AD HOC NETWORKING

IEEE Press
445 Hoes Lane
Piscataway, New Jersey

IEEE Press Editorial Board
Stamatios V. Kartalopoulos, *Editor in Chief*

M. Akay
J. B. Anderson
R. J. Baker
J. E. Brewer

M. E. El-Hawary
R. Leonardi
M. Montrose
M. S. Newman

F. M. B. Periera
C. Singh
S. Tewksbury
G. Zobrist

Kenneth Moore, *Director of Book and Information Services (BIS)*
Catherine Faduska, *Senior Acquisitions Editor*
Christina Kuhnen, *Associate Acquisitions Editor*

Technical Reviewers
Stephan Olariu, Old Dominion University, Norfolk, VA
Sergio Palazzo, Universita di Catania, Italy

MOBILE AD HOC NETWORKING

Edited by

STEFANO BASAGNI
Northeastern University

MARCO CONTI
Italian National Research Council (CNR)

SILVIA GIORDANO
University of Applied Science, Switzerland

IVAN STOJMENOVIC
University of Ottawa

IEEE

IEEE PRESS

WILEY-
INTERSCIENCE

A JOHN WILEY & SONS, INC., PUBLICATION

Copyright © 2004 by the Institute of Electrical and Electronics Engineers, Inc. All rights reserved.

Published simultaneously in Canada.

No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, except as permitted under Section 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 646-8600, or on the web at www.copyright.com. Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, (201) 748-6011, fax (201) 748-6008.

Limit of Liability/Disclaimer of Warranty: While the publisher and author have used their best efforts in preparing this book, they make no representation or warranties with respect to the accuracy or completeness of the contents of this book and specifically disclaim any implied warranties of merchantability or fitness for a particular purpose. No warranty may be created or extended by sales representatives or written sales materials. The advice and strategies contained herein may not be suitable for your situation. You should consult with a professional where appropriate. Neither the publisher nor author shall be liable for any loss of profit or any other commercial damages, including but not limited to special, incidental, consequential, or other damages.

For general information on our other products and services please contact our Customer Care Department within the U.S. at 877-762-2974, outside the U.S. at 317-572-3993 or fax 317-572-4002.

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print, however, may not be available in electronic format.

Library of Congress Cataloging-in-Publication Data is available.

ISBN 0-471-37313-3

Printed in the United States of America.

10 9 8 7 6 5 4 3 2 1

CONTENTS

CONTRIBUTORS

Giuseppe Anastasi received the Laurea (cum laude) degree in Electronics Engineering and Ph.D. in Computer Engineering, both from the University of Pisa, Italy, in 1990 and 1995, respectively. He is currently an associate professor of Computer Engineering at the Department of Information Engineering of the University of Pisa. His research interests include architectures and protocols for mobile computing, energy management, QoS in mobile networks, and ad hoc networks. He was a co-editor of the book, *Advanced Lectures in Networking,* and has published more than 40 papers, both in international journals and conference proceedings, in the area of computer networking. He served in the TPC of several international conferences including IFIP Networking 2002 and IEEE PerCom 2003. He is a member of the IEEE Computer Society.

Elizabeth M. Belding-Royer is an assistant professor in the Department of Computer Science at the University of California, Santa Barbara. She completed a Ph.D. in Electrical and Computer Engineering at University of California, Santa Barbara in 2000. Her research focuses on mobile networking, specifically routing protocols, security, scalability, and adaptability. Dr. Belding-Royer is the author of numerous papers related to ad hoc networking, has served on many program committees for networking conferences, and is currently the co-chair of the IRTF Ad Hoc Network Scalability (ANS) Research Group. She also sits on the editorial board for the Elsevier Science *Ad Hoc Networks Journal*. She is also the recipient of a 2002 Technology Review 100 award, presented to the world's top young investigators.

Amaresh Bikki received the Bachelor of Engineering with a major in Computer Science from Birla Institute of Technology and Sciences (BITS), Pilani, India in 1999. He then worked as a software engineer at Aditi Technologies, Bangalore, India before receiving a

Master Degree in Computer Science from the University of Texas, Dallas in 2002. He currently works in industry.

Luciano Bononi received the Laurea degree (summa cum laude) in Computer Science in 1997, and a Ph.D. in Computer Science in 2002, both from the University of Bologna, Italy. In 2000, he was a visiting researcher at the Department of Electrical Engineering of the University of California, Los Angeles. From March 2002 to September 2002, he was a postdoc researcher, and since October 2002, he has been a researcher at the Department of Computer Science of the University of Bologna. His research interests include wireless and mobile ad hoc networks, network protocols, power saving, modeling and simulation of wireless systems, discrete-event simulation, and parallel and distributed simulation.

Azzedine Boukerche is Canada Research chair and an associate professor of Computer Sciences at the School of Information Technology and Engineering (SITE), University of Ottawa, Canada. Prior to this, he was a faculty member in the Department of Computer Sciences, University of North Texas. He also worked as a senior scientist in the Simulation Sciences Division of Metron Corporation in San Diego. He spent the 1991–1992 academic year at Caltech/JPL where he contributed to a project centered about the specification and verification of the software used to control interplanetary spacecraft operated by Caltech/JPL–NASA Laboratory. His current research interests include ad hoc networks, mobile computing, wireless networks, parallel simulation, distributed computing, and large-scale distributed interactive simulation. Dr. Boukerche has published several research papers in these areas. He is the corecipient of the best research paper award at PADS'97, PADS'99, and MSWiM 2001. He has been general chair, program chair, and a member of the Program Committee of several international conferences and is an associate editor of the *International Journal of Parallel and Distributed Computing, SCS Transactions on Simulation, International Journal on Embedded Systems,* and a member of IEEE and ACM.

Raffaele Bruno received the Laurea degree in Telecommunications Engineering in 1999 and a Ph.D. in Information Engineering in 2003 from the University of Pisa, Italy. He is currently a junior researcher at the IIT Institute of the Italian National Research Council (CNR). From 2000 to 2002, he was honored with a fellowship from the Motorola R&D Center in Turin, Italy. His research interests are in the area of wireless and mobile networks with emphasis on efficient wireless MAC protocols, scheduling, and scatternet formation algorithms for Bluetooth networks.

Imrich Chlamtac holds a Ph.D. in Computer Science from the University of Minnesota. Since 1997, he has held the Distinguished Chair in Telecommunications at the University of Texas, Dallas and holds the titles of Sackler Professor at Tel Aviv University, Israel; Bruno Kessler Honorary Professor at the University of Trento, Italy; and University Professor at the Technical University of Budapest, Hungary. He also serves as president of Create-Net, an international research organization bringing together leading research institutes in Europe. Dr. Chlamtac is a Fellow of the IEEE and ACM societies, a Fulbright Scholar, and an IEEE Distinguished Lecturer. He is the winner of the 2001 ACM Sigmobile annual award, the IEEE ComSoc TCPC 2002 award for contributions to wireless and mobile networks, and multiple Best Paper awards in wireless and optical networks. Dr. Chlamtac has published more than 300 papers in refereed journals and conferences, and is the co-author of the first textbook on LANs, *Local Area Networks,* and *Mobile and Wireless Networks Protocols and Services* (Wiley, 2000). Dr. Chlamtac serves as the founding editor-in-chief of the ACM/URSI/Kluwer *Wireless Networks* (WINET) and the ACM/Kluwer *Mobile Networks and Applications* (MONET) journals, and the SPIE/Kluwer *Optical Networks Magazine (ONM).*

Scott Corson is vice president and chief network architect at Flarion Technologies, where he is responsible for the design of the IP network architecture enabled by the flash-ODFM air interface. Previously, he was on the faculty of the University of Maryland, College Park from 1995–2000, and was a consulting network architect for British Telecomm (BT) Labs, working on the design of an IP-based, fixed/cellular-converged network architecture from 1998–2000. He has worked on multiple access and network layer technologies for mobile wireless networks since 1987, and has been active in the Internet Engineering Task Force (IETF) since 1995. He co-organized and currently co-chairs the IETF Mobile Ad Hoc Networks Working Group, a body chartered to standardize mobile routing technology for IP-based networks of wireless routers. He has a Ph.D. in Electrical Engineering from the University of Maryland.

Sajal K. Das is a professor of Computer Science and Engineering and also the founding director of the Center for Research in Wireless Mobility and Networking (CReWMaN) at the University of Texas, Arlington (UTA). He is a recipient of UTA's Outstanding Faculty Research Award in Computer Science in 2001 and 2003, and the UTA College of Engineering Research Excellence Award in 2003. Dr. Das' current research interests include resource and mobility management in wireless networks, mobile and pervasive computing, wireless multimedia and QoS provisioning, sensor networks, mobile internet architectures and protocols, parallel processing, grid computing, performance modeling, and simulation. He has published more than 250 research papers in these areas, directed numerous industry and government funded projects, and holds four U.S. patents in wireless mobile networks. He received the Best Paper awards at ACM MobiCom'99, ICOIN'02, ACM MSWiM'00, and ACM/IEEE PADS'97. Dr. Das serves on the editorial boards of *IEEE Transactions on Mobile Computing,* ACM/Kluwer *Wireless Networks, Parallel Processing Letters,* and *Journal of Parallel Algorithms and Applications.* He served as general chair of IEEE PerCom 2004, MASCOTS'02, and ACM WoWMoM 2000-02; general vice chair of IEEE PerCom'03, ACM MobiCom'00, and IEEE HiPC'00-01; program chair of IWDC'02, WoWMoM'98-99; TPC vice chair of ICPADS'02; and as TPC member of numerous IEEE and ACM conferences. He is vice chair of the IEEE TCPP and TCCC executive committees and on the advisory boards of several cutting-edge companies.

András Faragó received a Bachelor of Science in 1976, Master of Science in 1979, and Ph.D. in 1981, all in Electrical Engineering from the Technical University of Budapest, Hungary. After graduation, he joined the Department of Mathematics, Technical University of Budapest and in 1982 he moved to the Department of Telecommunications and Telematics. He was also cofounder and research director of the High Speed Networks Laboratory, the first research center in high-speed networking in Hungary. In 1996, he was honored the distinguished title "Doctor of the Hungarian Academy of Sciences." In 1998, he joined the University of Texas, Dallas as professor of Computer Science. Dr. Farago has authored more than 100 research papers and his work is currently supported by three research grants from the National Science Foundation. His main research interest is in the development and analysis of algorithms, network protocols, and modeling of communication networks.

Laura Marie Feeney has been a member of the Computer and Network Architecture Laboratory at the Swedish Institute of Computer Science in Kista, Sweden since 1999. Her research includes topics in energy efficiency, routing, and quality of service for wireless networks, especially ad hoc and sensor networks. Much of her work is related to problems in cross-layer interaction. She also participated in the development of SpontNet, a prototype platform for studying service architectures for secure, application-specific ad hoc networks created among a small group of users. She is also an occasional guest lecturer for networking courses at Sweden's Royal Institute of Technology and Luleaa University of Technology. Ms. Feeney's research interests include many topics in systems and networking and she has an especially strong interest in experimenting with real systems and in combining analytic models, simulation, and measurement. She is a member of the ACM.

Enrico Gregori received the Laurea degree in Electronic Engineering from the University of Pisa in 1980. In 1981, he joined the Italian National Research Council (CNR) where he is currently the deputy director of the CNR Institute for Informatics and Telematics (IIT). In 1986, he held a visiting position in the IBM research center in Zurich, working on network software engineering and heterogeneous networking. He has contributed to several national and international projects on computer networking. He has authored more than 100 papers in the area of computer networks, has published in international journals and conference proceedings, and is co-author of the book, *Metropolitan Area Networks*. He was the general chair of the IFIP TC6 conferences Networking2002 and PWC2003 (Personal Wireless Communications). He served as guest editor for the Networking2002 journal special issues on Performance Evaluation and Cluster Computing the ACM/Kluwer *Wireless Networks Journal*. He is a member of the board of directors of the Create-Net Association, an association of several Universities and research centers which foster research on networking at the European level. He is on the editorial board of the *Cluster Computing* and the *Computer Networks Journal*. His current research interests include ad hoc networks, sensor networks, wireless LANs, quality of service in packet-switching networks, and evolution of TCP/IP protocols.

Xiang-Yang Li has been an assistant professor of Computer Science at the Illinois Institute of Technology since August 2000. He joined the Computer Science Department of University of Illinois at Urbana–Champaign in 1997 and received the Master of Science and Ph.D. in Computer Science in 2000 and 2001. Since 1996, his research interests span computational geometry, wireless ad hoc networks, optical networks, and algorithmic mechanism design. Since 1998, he has authored or co-authored five book chapters, 20 journal papers, and more than 40 conference papers in the areas of computational geometry, wireless networks, and optical networks. He won the Hao Wang award at the 7th Annual International Computing and Combinatorics Conference (COCOON). He is a member of IEEE and ACM.

Jennifer J-N. Liu has more than 10 years of broad new technology and networking protocol development experience in the telecommunication industry. Ms. Liu started her career in 1993 as a member of scientific staff at Nortel's Bell–Northern Research, developing platforms for the next-generation DMS switch. In 1997, she joined Alcatel's Motorola Division and participated in designing signaling and call-processing software components for Motorola's EMX CDMA switch. She became part of the initial IP Connection management team in 1998 that started Alcatel's VoIP SoftSwitch A1000 CallServer project, and later led the development for the IP Sigtran protocols/applications. Since 2000, she has worked in startups, and has helped in creating MPLS/RSVP-based network traffic/bandwidth management strategies and QoS solutions for Metera Networks, as well as VoIP related services/gateway management features for Westwave Communications. Ms. Liu is an inventor/co-inventor of several patents in the networking field. She received a Master of Science from the Department of Systems and Computer Engineering at Carleton University in Ottawa, Canada. She is currently doing Ph.D. studies in the Department of Computer Science at the University of Texas, Dallas.

Joseph P. Macker is a senior communication systems and network research scientist at the Naval Research Laboratory in Washington, D.C. Currently, he leads the Protocol Engineering and Advanced Networking (Protean) Group that is investigating adaptive networking solutions for both mobile wireless and wired networking architectures. He holds a Master of Science from George Washington University in Communications Theory and a Bachelor of Science from the University of Maryland, College Park. His primary research interests are adaptive network protocol and architecture design, multicast technology and data reliability, mobile wireless networking and routing, network protocol simulation and analysis, Quality of Service (QoS) networking, multimedia networking, and adaptive sensor networking. Mr. Macker has served as co-chairman of the Mobile Ad Hoc Networking (MANET) Working Group within the Internet Engineering Task Force (IETF). He has also served on the Steering and Program committees for the annual ACM Mobihoc Symposium events. His present work focuses on dynamic, ad hoc networking technology and its application to wireless communication and sensor networks.

Pietro Michiardi received the Laurea degree in Electronic Engineering from the Politecnico di Torino in 2001. He was granted a scholarship by the European Union to take part in a program in advanced telecommunications engineering at the Eurecom Institute, where he got a diploma in Multimedia Communications. In January 2000, Mr. Michiardi joined the Eurecom Institute as a research engineer working on a project for the development of advanced security services for business transactions. Since September 2001, Pietro has been a Ph.D. student at the Eurecom Institute, working on routing security and cooperation enforcement for mobile ad hoc networks. Pietro Michiardi contributed actively to the definition of new types of security requirements for the ad hoc network paradigm and proposed original security mechanisms that were analyzed using economic principles. His work on the use of game theory to model cooperation in ad hoc networks and to study cooperation-enforcement mechanisms was awarded in the IEEE/ACM WiOpt 2003 International Workshop on Modeling and Optimization for Wireless Networks.

Refik Molva has been a professor at Institut Eurécom since 1992. He leads the network security research group that currently focuses on multipoint security protocols, multicomponent system security, and security in ad hoc networks. His past projects at Eurécom were on mobile code protection, mobile network security, anonymity, and intrusion detection. Beside security, he worked on distributed multimedia applications and was responsible for the BETEUS European project on CSCW over a trans-European ATM network. Prior to joining Eurécom, he worked for five years as a research staff member in the Zurich Research Laboratory of IBM, where he was one of the key designers of the KryptoKnight security system. He also worked as a network security consultant in the IBM Consulting Group in 1997. He is the author of several publications and patents in the area of network security and has been part of several evaluation committees for various national and international bodies, including the European Commission.

Chiara Petrioli received the Laurea degree with honors in Computer Science in 1993, and a Ph.D. in Computer Engineering in 1998, both from Rome University "La Sapienza," Italy. She is currently assistant professor at the Computer Science Department at La Sapienza, The University of Rome. Her current work focuses on ad hoc and sensor networks, Bluetooth, energy-conserving protocols, QoS in IP networks, and content delivery networks. Prior to Rome University, she was research associate at Politecnico di Milano, and was working with the Italian Space Agency (ASI) and Alenia Spazio. Dr. Petrioli is the author of several papers in the areas of mobile communications and IP networks, is an area editor of the ACM *Wireless Networks Journal,* of the Wiley *Wireless Communications and Mobile Computing Journal,* and of the Elsevier *Ad Hoc Networks Journal.* She has served on the organizing committee and technical program committee of several leading conferences in the area of networking and mobile computing, including ACM Mobicom, ACM MobiHoc, and IEEE ICC.

Ram Ramanathan is a division scientist at BBN Technologies. His research interests are in the area of wireless and ad hoc networks, in partcular, routing, medium-access control, and directional antennas. He is currently the principal investigator for a project on architecture and protocols for opportunistic access of spectrum using cognitive radios. Recently, he was one of one of two principal investigators for the DARPA project UDAAN (Utilizing Directional Antennas for Ad Hoc Networking) and the co-investigator on NASA's Distributed Spacecraft Network project. Ram is actively involved in the evolution of mobile ad hoc networking, and has recently served on the program and steering committees of the ACM MobiHoc Symposium and ACM Mobicom. He is on the editorial board of *Ad Hoc Networks* journal. He has won three Best Paper awards at prestigious conferences—ACM Sigcomm 92, IEEE Infocom 96, and IEEE Milcom 02. Dr. Ramanathan holds a Bachelor of Technololgy from the Indian Institute of Technology, Madras, and a Master of Science and a Ph.D. from the University of Delaware. He is a senior member of the IEEE.

Andreas Savvides received a Bachelor of Science in Computer Engineering from the University of California, San Diego in 1997, a Master of Science in Computer Engineering from the University of Massachusetts, Amherst in 1999, and a Ph.D. in Electrical Engineering from the University of California, Los Angeles in 2003. He is currently an assistant professor in Electrical Engineering and Computer Science at Yale University. In 1999, Andreas also worked in ad hoc networking at the HRL Labs in Malibu, California. His research interests are in sensor networks, embedded systems, and ubiquitous computing. He is a member of IEEE and ACM.

Mani Srivastava received a Bachelor of Technology in 1985 from IIT Kanpur, a Master of Science in 1987 and Ph.D. in 1992 from the University of California, Berkeley and is a professor of electrical Engineering at UCLA, where he directs the Networked and Embedded Systems Laboratory and is associated with the Center for Embedded Networked Sensing. Prior to joining UCLA, he was at Bell Laboratories Research, Murray Hill. His current research spans all aspects of wireless, embedded, and low-power systems, with a particular focus on systems issues and applications in wireless sensor and actuator networks. The research in his group is funded by DARPA, ONR, NSF, and the SRC. He has published more than 100 papers, is a co-inventor on five U.S. patents in mobile and wireless systems, and has served on the editorial boards and program committees of leading journals and conferences in his field. His work has been recognized by awards such as the President of India's Gold Medal (1985), Best Paper award at the IEEE ICDCS (1997), the NSF Career Award (1997), the Okawa Foundation Grant (1998), and the second prize at the ACM DAC Design Contest (2002).

Violet R. Syrotiuk is an assistant professor of Computer Science and Engineering at Arizona State University. Her research interests include many aspects of medium-access control for mobile ad hoc networks, such as dynamic adaptation, quality of service, energy awareness, and topology transparency. She also has an interest in design and analysis of experiments for identifying protocol interactions, and the use of formal modeling and optimization for improved cross-layer designs. Dr. Syrotiuk's research is currently supported by three grants from the National Science Foundation and by the DARPA Connectionless Networks program. In the past, her work has been supported by the DARPA Next Generation (XG), Future Combat Systems (FCS), and Globile Mobile Information (GloMo) programs. She serves on the Technical Program and Organizing committees of major conferences in mobile networking and computing and is a member of the ACM and IEEE.

Alessandro Urpi received a Bachelor of Science in Computer Science from the University of Pisa. He is currently a third-year Ph.D. student in the Computer Science Department, University of Pisa. His interests include wireless networking modeling, protocols, and algorithms for ad hoc networks, switching, and switch architectures. In these areas, he published some conference and journal papers, and won the "Best Student Paper Award" at Networking 2002. His Ph.D. thesis addresses cooperation analysis in wireless mobile ad hoc networks.

Jie Wu a professor in the Department of Computer Science and Engineering, Florida Atlantic University. He has published more than 200 papers in various journals and conference proceedings. His research interests are in the area of mobile computing, routing protocols, fault-tolerant computing, and interconnection networks. Dr. Wu served as a program vice chair for the 2000 International Conference on Parallel Processing (ICPP) and a program vice chair for 2001 IEEE International Conference on Distributed Computing Systems (ICDCS). He was a program co-chair of the 12th ISCA International Conference on Parallel and Distributed Computing Systems in 1999. He is the author of the text, *Distributed System Design*. Currently, Dr. Wu serves as an Associate Editor of *IEEE Transactions on Parallel and Distributed Systems (TPDS)* and four other international journals. He also served as a guest editor of *IEEE TPDS, Journal of Parallel and Distributed Computing (JPDC),* and *IEEE Computer.* Dr. Wu was a recipient of the 1996–1997 and 2001–2002 Researcher of the Year Award at Florida Atlantic University. He is also a recipient of the 1998 Outstanding Achievements Award from IASTED. He served as an IEEE Computer Society Distinguished Visitor. Dr. Wu is a member of ACM and a senior member of IEEE.

Gergely V. Záruba is an assistant professor of Computer Science and Engineering at The University of Texas, Arlington. He received a Ph.D. degree in Computer Science from The University of Texas, Dallas in 2001, and his Master of Science in Computer Engineering from the Technical University of Budapest, Department of Telecommunications and Telematics, in 1997. He is a member of the Center for Research in Wireless Mobility and Networking (CReWMaN). Dr. Zaruba's research interests include wireless networks, algorithms, and protocols, and performance evaluation concentrating on the medium-access control layer and current wireless technologies. He has served on many organizing and technical program committees for leading conferences and has guest edited an ACM *MONET* journal on research related to the Bluetooth technology. He is a member of the IEEE and ACM.

PREFACE

Whereas today's expensive wireless infrastructure depends on centrally deployed hub-and-spoke networks, mobile ad hoc networks consist of devices that are autonomously self-organizing in networks. In ad hoc networks, the devices themselves are the network, and this allows seamless communication, at low cost, in a self-organized fashion and with easy deployment. The large degree of freedom and the self-organizing capabilities make mobile ad hoc networks completely different from any other networking solution. For the first time, users have the opportunity to create their own network, which can be deployed easily and cheaply. However, a price for all those features is paid in terms of complex technology solutions, which are needed at all layers and also across several layers.

For all those reasons, mobile ad hoc networking is one of the more innovative and challenging areas of wireless networking, and this technology promises to become increasingly present in everybody's life. Ad hoc networks are a key step in the evolution of wireless networks. They inherit the traditional problems of wireless and mobile communications, such as bandwidth optimization, power control and transmission quality enhancement. In addition, the multihop nature and the lack of fixed infrastructure brings new research problems such as network configuration, device discovery and topology maintenance, as well as ad hoc addressing and self-routing. Many different approaches and protocols have been proposed and there are multiple standardization efforts within the Internet Engineering Task Force and the Internet Research Task Force, as well as academic and industrial projects.

This book is the result of our effort to put together a representative collection of chapters covering the most advanced research and development in mobile ad hoc networks. It is based on a number of stand-alone chapters that are deeply interconnected. It seeks to provide an opportunity for readers to find advances on a specific topic, as well as to explore the whole field of rapidly emerging mobile ad hoc networks. In addition, the historical evolution and the role of mobile ad hoc networks in 4G mobile systems are discussed in depth in the first chapter.

In most of the past research, mobile ad hoc networks are seen as part of the Internet, with IP-centric layered architecture. This architecture has two main advantages: it simplifies the interconnection to the Internet, and guarantees the independence from (heterogeneous) wireless technologies. The layered paradigm, which has significantly simplified the Internet design and led to the robust scalable protocols, can result in poor performances when applied to mobile ad hoc networks. In fact, in mobile ad hoc networks several functions can hardly be isolated into a single layer. Energy management, security and cooperation, quality of service, among the others, cannot be completely confined in a unique layer. Rather, their implementation results are more effective by exploiting and interacting with mechanisms at all layers. A more efficient and performing architecture for mobile ad hoc networks thus should avoid a strict layering approach, but rather follow an integrated and hierarchical framework to take advantage of the interdependencies among layers. This book goes in this new direction by presenting *cross-layering* chapters. Most of the chapters do not focus on single-layer mechanisms, rather they present and discuss functions that are implemented by combining mechanisms that, in a strict layered architecture, belong to different layers.

Inside the ad hoc networking field, wireless sensor networks play a special role, as they are used mainly for phenomena monitoring. The solutions for mobile ad hoc networks are rarely suitable for sensor networks, as the latter are rarely mobile in a strict sense, and prone to different constraints deriving by the sensing devices' features and by application requirements. This generated an extensive literature that could hardly be accommodated in this book without being reductive.

This book is intended for developers, researchers, and graduate students in computer science and electrical engineering, as well as researchers and developers in the telecommunication industry. The editors of this book first discussed the selection of problems and topics to be covered and then discussed the choice of best authors for each of the selected topics. We believe that we have achieved a balanced selection of chapters with top quality experts selected for presenting the state of the art on each topic. The editors envision the introduction of a number of computer science and electrical engineering graduate courses in ad hoc networks, and believe that this book provides textbook quality for use in such courses.

The editors are particularly grateful to the authors who have agreed to present their work in this book. They would also like to express their sincere thanks to all the reviewers, whose helpful remarks have contributed to the outstanding quality of this book. Special thanks go to Stephen Olariu and Sergio Palazzo; we have benefited enormously from their comments and suggestions. Finally, we are immensely grateful to Catherine Faduska and Christina Kuhnen for their invaluable collaboration in putting this book together.

STEFANO BASAGNI
MARCO CONTI
SILVIA GIORDANO
IVAN STOJMENOVIC

April 2004

CHAPTER 1

MOBILE AD HOC NETWORKING WITH A VIEW OF 4G WIRELESS: IMPERATIVES AND CHALLENGES

JENNIFER J.-N. LIU and IMRICH CHLAMTAC

1.1 INTRODUCTION

The wireless arena has been experiencing exponential growth in the past decade. We have seen great advances in network infrastructures, growing availability of wireless applications, and the emergence of omnipresent wireless devices such as portable or handheld computers, PDAs, and cell phones, all getting more powerful in their capabilities. These devices are now playing an ever-increasingly important role in our lives. To mention only a few examples, mobile users can rely on their cellular phone to check e-mail and browse the Internet; travelers with portable computers can surf the internet from airports, railway stations, cafes, and other public locations; tourists can use GPS terminals installed inside rental cars to view driving maps and locate tourist attractions; files or other information can be exchanged by connecting portable computers via wireless LANs while attending conferences; and at home, a family can synchronize data and transfer files between portable devices and desktops.

Not only are mobile devices getting smaller, cheaper, more convenient, and more powerful, they also run more applications and network services. All of these factors are fueling the explosive growth of the mobile computing equipment market seen today. Market reports from independent sources show that the worldwide number of cellular users has been doubling every 1½ years, with the total number growing from 23 million in 1992 to 860 million in June 2002. This growth is being fueled further by the exploding number of

Mobile Ad Hoc Networking. Edited by Basagni, Conti, Giordano, and Stojmenovic.
ISBN 0-471-37313-3 © 2004 Institute of Electrical and Electronics Engineers, Inc.

Internet and laptop users [6]. Projections show that in the next two years, the number of mobile connections and the number of shipments of mobile and Internet terminals will grow by yet by another 20–50% [6]. With this trend, we can expect the total number of mobile Internet users soon to exceed that of fixed-line Internet users.

Among the myriad of applications and services run by mobile devices, network connections and corresponding data services are without doubt in highest demand. According to a recent study by Cahners In-Stat Group, the number of subscribers to wireless data services will grow rapidly from 170 million worldwide in 2000 to more than 1.3 billion in 2004, and the number of wireless messages sent per month will rise dramatically from 3 billion in December 1999 to 244 billion by December 2004. Currently, most of the connections among wireless devices occur over fixed-infrastructure-based service providers or private networks; for example, connections between two cell phones set up by BSC and MSC in cellular networks, or laptops connected to the Internet via wireless access points. Although infrastructure-based networks provide a great way for mobile devices to get network services, it takes time to set up the infrastructure network, and the costs associated with installing infrastructure can be quite high. There are, furthermore, situations in which user-required infrastructure is not available, cannot be installed, or cannot be installed in time in a given geographic area. Providing the needed connectivity and network services in these situations requires a mobile ad hoc network.

For all of these reasons, combined with significant advances in technology and standardization, new alternative ways to deliver connectivity have been gaining increased attention in recent years. These are focused around having mobile devices within the transmission range connect to each other through automatic configuration, setting up an ad hoc mobile network that is both flexible and powerful. In this way, not only can mobile nodes communicate with each other, but also receive Internet services through an Internet gateway node, effectively extending both network and Internet services to noninfrastructure areas. As the wireless network continues to evolve, this ad hoc capability will become more important, and the technology solutions used to support it more critical, spurring a host of research and development projects and activities in industry and academia alike.

This chapter dwells on the impetus behind the inevitable market adoption of the mobile ad hoc network, and presents a representative collection of technology solutions that can be used in different layers of the network, especially the algorithms and protocols needed for its operation and configuration. In the following section, we review the wireless communication technologies, the types of wireless networks and their evolution path, as well as the problems and market demands for existing wireless systems. We then explain why ad hoc networking is expected to form the essential piece in the 4G network architecture. In Section 1.3, we look at the mobile ad hoc network in closer detail, covering its specific characteristics, advantages, and design challenges. After that, we show the range of opportunities for MANET applications, both military and commercial, which also serve to elaborate the market potential behind MANET technology advancement. Section 1.4 summarizes the current status and design challenges facing the research community. A large number of protocols and algorithms have been developed for mobile ad hoc networks, which are presented, discussed and compared in Section 1.4. Although impressive research and development results are demonstrated in this and the remaining detailed chapters in this book, many open issues remain to clear the path for the successful ad hoc network deployment and commercialization. Some of the open research problems in ad hoc wireless networking are the subject of Section 1.5. Section 1.6 presents conclusions, and introduces the rest of chapters in this book.

1.2 REVIEW OF WIRELESS NETWORK EVOLUTION

The wireless communication landscape has been changing dramatically, driven by the rapid advances in wireless technologies and the greater selection of new wireless services and applications. The emerging third-generation cellular networks have greatly improved data transmission speed, which enables a variety of higher-speed mobile data services. Meanwhile, new standards for short-range radio such as Bluetooth, 802.11, Hiperlan, and infrared transmission are helping to create a wide range of new applications for enterprise and home networking, enabling wireless broadband multimedia and data communication in the office and home.

Before delving into these technologies and applications, we first examine some of the main characteristics of wireless communication as related to specification and classification of these networks, and then review the key capabilities exhibited by the various types of wireless networks.

1.2.1 Wireless Communication Characteristics

In general, wireless networking refers to the use of infrared or radio frequency signals to share information and resources between devices. Many types of wireless devices are available today; for example, mobile terminals, pocket size PCs, hand-held PCs, laptops, cellular phone, PDAs, wireless sensors, and satellite receivers, among others.

Due to the differences found in the physical layer of these systems, wireless devices and networks show distinct characteristics from their wireline counterparts, specifically,

- Higher interference results in lower reliability.
  - —Infrared signals suffer interference from sunlight and heat sources, and can be shielded/absorbed by various objects and materials. Radio signals usually are less prone to being blocked; however, they can be interfered with by other electrical devices.
  - —The broadcast nature of transmission means all devices are potentially interfering with each other.
  - —Self-interference due to multipath.
- Low bandwidth availability and much lower transmission rates, typically much slower-speed compared to wireline networks, causing degraded quality of service, including higher jitter, delays, and longer connection setup times.
- Highly variable network conditions:
  - —Higher data loss rates due to interference
  - —User movement causes frequent disconnection
  - —Channel changes as users move around
  - —Received power diminishes with distance
- Limited computing and energy resources: limited computing power, memory, and disk size due to limited battery capacity, as well as limitation on device size, weight, and cost.
- Limited service coverage. Due to device, distance, and network condition limitations, service implementation for wireless devices and networks faces many constraints and is more challenging compared to wired networks and elements.

- Limited transmission resources:
  —Medium sharing
  —Limited availability of frequencies with restrictive regulations
  —Spectrum scarce and expensive
- Device size limitation due to portability requirements results in limited user interfaces and displays.
- Weaker security: because the radio interface is accessible to everyone, network security is more difficult to implement, as attackers can interface more easily.

1.2.2. Types of Wireless Networks

Many types of wireless networks exist, and can be categorized in various ways set out in the following subsections depending on the criteria chosen for their classification.

1.2.2.1. By Network Formation and Architecture. Wireless networks can be divided into two broad categories based on how the network is constructed and the underlining network architecture:

1. Infrastructure-based network. A network with preconstructed infrastructure that is made of fixed and wired network nodes and gateways, with, typically, network services delivered via these preconfigured infrastructures. For example, cellular networks are infrastructure-based networks built from PSTN backbone switches, MSCs, base stations, and mobile hosts. Each node has its specific responsibility in the network, and connection establishment follows a strict signaling sequence among the nodes [2]. WLANs typically also fall into this category.
2. Infrastructureless (ad hoc) network. In this case a network is formed dynamically through the cooperation of an arbitrary set of independent nodes. There is no prearrangement regarding the specific role each node should assume. Instead, each node makes its decision independently, based on the network situation, without using a preexisting network infrastructure. For example, two PCs equipped with wireless adapter cards can set up an independent network whenever they are within range of one another. In mobile ad hoc networks, nodes are expected to behave as routers and take part in discovery and maintenance of routes to other nodes.

1.2.2.2. By Communication Coverage Area. As with wired networks, wireless networks can be classified into different types based on the distances over which data is transmitted:

1. Wireless Wide Area Networks (Wireless WANs). Wireless WANs are infrastructure-based networks that rely on networking infrastructures like MSCs and base stations to enable mobile users to establish wireless connections over remote public or private networks [3]. These connections can be made over large geographical areas, across cities or even countries, through the use of multiple antenna sites or satellite systems maintained by wireless service providers. Cellular networks (like GSM networks or CDMA networks) and satellite networks are good examples of wireless WAN networks.

2. Wireless Metropolitan Area Networks (Wireless MANs). Wireless MAN networks are sometimes referred to as fixed wireless. These are also infrastructure-based networks that enable users to establish broadband wireless connections among multiple locations within a metropolitan area, for example, among multiple office buildings in a city or on a university campus, without the high cost of laying fiber or copper cabling and leasing lines [3]. In addition, Wireless MANs can serve as backups for wired networks should the primary leased lines for wired networks become unavailable. Both radio waves and infrared light can be used in wireless MANs to transmit data. Popular technologies include local multipoint distribution services (LMDS) and multichannel multipoint distribution services (MMDS). IEEE has set up a specific 802.16 Working Group on Broadband Wireless Access Standards that develops standards and recommended practices to support the development and deployment of broadband wireless metropolitan area networks [151].
3. Wireless Local Area Network (Wireless LANs). Wireless local area networks enable users to establish wireless connections within a local area, typically within a corporate or campus building, or in a public space, such as an airport, usually within a 100 m range. WLANs provide flexible data communication systems that can be used in temporary offices or other spaces where the installation of extensive cabling would be prohibitive, or to supplement an existing LAN so that users can work at different locations within a building at different times [3, 7]. Offices, homes, coffee shops, and airports represent the typical hotspots for wireless LAN installations.

   Wireless LANs can operate in infrastructure-based or in ad hoc mode. In the infrastructure mode, wireless stations connect to wireless access points that function as bridges between the stations and an existing network backbone. In the ad hoc mode, several wireless stations within a limited area, such as a conference room, can form a temporary network without using access points, if they do not require access to network resources.

   Typical wireless LAN implementations include 802.11 (Wi-Fi) and Hiperlan2. Under 802.11a and 802.11b, data can reach transmission speeds between 11 Mbps to 54 Mbps [13, 14].
4. Wireless Personal Area Networks (Wireless PANs). Wireless PAN technologies enable users to establish ad hoc, wireless communication among personal wireless devices such as PDAs, cellular phones, or laptops that are used within a personal operating space, typically up to a 10 meter range. Two key Wireless PAN technologies are Bluetooth and infrared light. Bluetooth [10, 11] is a cable-replacement technology that uses radio waves to transmit data to a distance of up to 9–10 m, whereas infrared can connect devices within a 1 m range. Wireless PAN is gaining momentum because of its low complexity, low power consumption, and interoperability with 802.11 networks.

1.2.2.3. By Access Technology. Depending on the specific standard, frequency, and spectrum usage, wireless networks can be categorized based on the access technology used. These include:

- GSM networks
- TDMA networks
- CDMA networks

- Satellite networks
- Wi-Fi (802.11) networks
- Hiperlan2 networks
- Bluetooth networks
- Infrared networks

1.2.2.4. By Network Applications. Wireless networks can also be categorized based on the specific usage and applications they support, for example,

1. Enterprise Networks
2. Home Networks
3. Tactical Networks
4. Sensor Networks
5. Pervasive Networks
6. Wearable Computing
7. Automated Vehicle Networks

1.2.3. Forces Driving Wireless Technology Evolution

To understand the wireless technology trends, and to see why noninfrastructure-based mobile ad hoc networks are poised to play an important role in the evolution of future wireless networks, it helps to review the evolution path of different technology generations. Table 1.1 summarizes the technologies, architectures, and applications for each of these generations.

One can argue that the commercial history of wireless started with the first generation or 1G in 1980s, which supported analog cell phones using FDMA and was relatively unsophisticated. Because different regions of the world pursued different mobile phone standards, 1G phones typically could only be used within one country. E Examples of 1G systems include NMT, TACS in Europe, and AMPS in North America.

The cellular industry began deployment of second-generation networks, 2G, a decade or so ago. 2G digitizes the mobile system and adds fax, data, and messaging capabilities on top of the traditional voice service. This evolution was triggered by the high demand for low-speed data access required to enable popular mobile data services like email, SMS, and so on. Again, different standards were deployed in different regions of the world; for example, Europe and Asia use GSM, whereas North America uses a mix of TDMA, CDMA, and GSM as 2G technologies. Recently, 2G has been extended to 2.5G to provide better support for transmitting low-speed data up to 384 kbps.

Currently, efforts are under way to transition the wireless industry from 2G networks to third-generation (3G) networks that would follow a common global standard based on CDMA and provide worldwide roaming capabilities. 3G networks offer increased bandwidth of 128 Kbps when mobile device is moving at higher speeds, for example, a car, up to 384 Kbps for mobility at pedestrian speed, and 2 Mbps in stationary applications, making it possible to deliver live video clips. There are still different flavors of the air interfaces though: Europe and Asia are promoting W-CDMA and EDGE, whereas North America works on cdma2000, each developed by different standard bodies—3GPP for Europe and Asia and 3GPP2 for North America.

Table 1.1. Wireless Technology Generations

Generation	1G	2G	2.5G	3G	4/5G
Time Frame	1980s	1990s	Late1990s	2000s (2010 full deployment)	2010s
Signal Type	Analog	Digital	Digital	Digital	Digital
Access					
Multiple access	FDMA/FDD	TDMA/FDD CDMA/FDD	EDGE, GPRS	CDMA, W-CDMA, TD-SCDMA	MC-CDMA, OFDM
Frequency spectrum		824–894 MHz 890–960 MHz 1850–1990 MHz (PCS)		1800–2400 MHZ (varies country to country)	Higher-frequency bands 2–8 GHz
Bandwidth				5–20 Mhz	≥100 MHz
Antenna				Optimized antenna, multiband adapter	Smarter antenna, Multiband and wide-band support
FEC				Convolutional rate, 1/2, 1/3	Concatenated coding scheme
Network Architecture					
Media type	Voice	Mostly voice Low-speed data services via modem (10–70 kbps)	Mostly Voice Higher-speed data (10–384 kbps)	Voice High-speed data (144 kbps–2 Mbps)	Converged voice/data/multimedia over IP; Ultra-high-speed data (2–100 Mbps)
Network type	Cellular	Cellular	Cellular	WWAN Cell based	Integrated WWAN, WMAN, WLAN (Wi-Fi, Bluetooth) and WPAN (Bluetooth)
Structure	Infrastructure based	Infrastructure based	Infrastructure based	Infrastructure-based network	Hybrid of infrastucture-based and ad hoc network
Switching	Circuit switched	Circuit switched	Circuit switched	Circuit switched and packet switched	Packet switched
IP support	N/A	N/A	N/A	Use several air link protocols, including IP5.0	All IP based (IP6.0)
New Applications				Emails, maps/directions, News, shopping, e-commerce, interactive gaming, etc.	Ubiquitous computing with location intelligence
Ex System	AMPS, NMT, TACS	GSM, DCS1900, IS-136, CdmaOne	GPRS, EDGE	UMTS IMT2000 CDMA2000 W-CDMA	

Despite the high expectations for 3G networks, 3G is facing difficulties getting deployed and meeting its promised performance and throughput due to architecture and capability limitations. On the other hand, recent technology advancements enable new services and thus impose new requirements on system capabilities that were not taken into consideration in the original 3G system design. Let us take a closer look at some of these.

1.2.3.1. The need to integrate various types of wireless networks. Today's wireless communication systems are primarily designed to provide cost-efficient wide-area coverage for users with moderate bandwidth demands, and 3G is based on primarily a wide-area concept. Many other types of wireless networks have since been designed and are gaining popularity, including wireless LAN and PAN networks, but these are being designed as logically separate networks. The various wireless networks need to be integrated in order to provide seamless wireless services. Emerging technology trends indicate that future-generation communication systems will consist of a high-speed wired backbone and wireless local area networks attached to the periphery of the network. Wireless LANs and PANs will extend the coverage of broadband services and provide ubiquitous network access to mobile users [172].

1.2.3.2. The need to integrate wireless platforms with fixed network backbone infrastructures. The consumer of telecommunication services of tomorrow will expect to receive the same services in a wireless fashion as he receives from a fixed network. A wireless system should, therefore, be transparent to the user and thus highly integrated with the fixed network backbone like Internet and PSTN networks.

1.2.3.3. The need to support high-speed multimedia services. Growth in Internet information services and the emergence of new multimedia applications including music, video streaming, or videoconferencing make multimedia services highly attractive to wireless users. In 3G systems, the maximum data speed supported is 2 Mbit/s, bandwidth that is not sufficient to meet the needs of these high-performance applications [143]. Very high-speed data transmission speed has to be supported in order to enable multimedia services on mobile devices.

1.2.3.4. The need for convergence in network infrastructure. Today, wireless communications are heavily biased toward voice. With data traffic growing at almost exponential speed and IP becoming prevalent, maintaining two separate backbone infrastructure for voice and data traffic becomes untenable. Converged IP-based digital packet networks that can support voice, data, as well as multimedia applications at the same time provide the ideal platform to lower network operating cost and enable new breeds of network services [15].

1.2.3.5. The need to support high mobility and device portability. High mobility and device portability enable wireless users to connect to networks and communicate with other users or devices anytime, anywhere [12]. 3G systems cannot yet fully support this transparency, such as dynamically changing network addresses and device locations. Progress is needed to eliminate the shortcomings of wireless systems so that the inherent convenience of mobility will no longer cause deterioration of system functionalities.

Another aspect of portability relates to the need to make the mobile device more usable, to extend the battery power, make devices smaller, and create better user interfaces that match the conventional environment.

1.2.3.6. The need to support noninfrastructure-based networks. Current wireless systems rely on preconfigured infrastructure (routers, MSCs, base stations) to deliver wireless services. This limits the service availability in established areas. However, in many situations networking services are required where infrastructure is not available or not deliverable in a short period of time, for example, in combat or emergency situations. Support and integration of noninfrastructure-based networks becomes important in these situations.

1.2.3.7. The need to add location intelligence. As adoption of mobile wireless systems continues to grow, wireless users will demand services that utilize the convenience stemming from mobility. Among these services, location-based information services, such as getting driving or service directions, location-dependent query support, and system configuration are becoming commonplace and need to be gradually added to system capabilities.

1.2.3.8. The need to lower the cost of wireless services. Cost is one of the key nontechnical issues that need to be dealt with in 3G systems. For example, the cost is exceedingly high for deployment. 3G spectrum licenses are auctioned at very high prices of more than $100 billion. Being able to lower these costs while providing better services is a key requirement for future network success. One of the ways to lower the infrastructure cost is, for example, by successfully implementing convergence of voice and multimedia into IP networks.

1.2.3.9. The need for greater standard interoperability. Multiple air interface standards in 3G are making it difficult for devices to roam and interoperate across networks. Furthermore, global mobility and service portability cannot be fully achieved without universal network standardization. Areas needing additional standardization start from lower-layer issues such as modulation techniques, spectrum allocation, and signaling, and continue all the way up to protocols and enabling architectures discussed in the remainder of this chapter.

To meet these new requirements and overcome the limitations and problems of current 3G systems, new architectures and capabilities need to be incorporated into the next-generation wireless systems to provide the much needed improvements.

1.2.4. 4G Wireless Architecture and Capabilities

4G is all about an integrated global network based on an open-systems approach. Integrating different types of wireless networks with wireline backbone networks seamlessly and the convergence of voice, multimedia, and data traffic over a single IP-based core network will be the main focus of 4G. With the availability of ultrahigh bandwidth of up to 100 Mbps, multimedia services can be supported efficiently. Ubiquitous computing is enabled with enhanced system mobility and portability support, and location-based services and support of ad hoc networking are expected. Figure 1.1 illustrates the networks and components within the 4G network architecture.

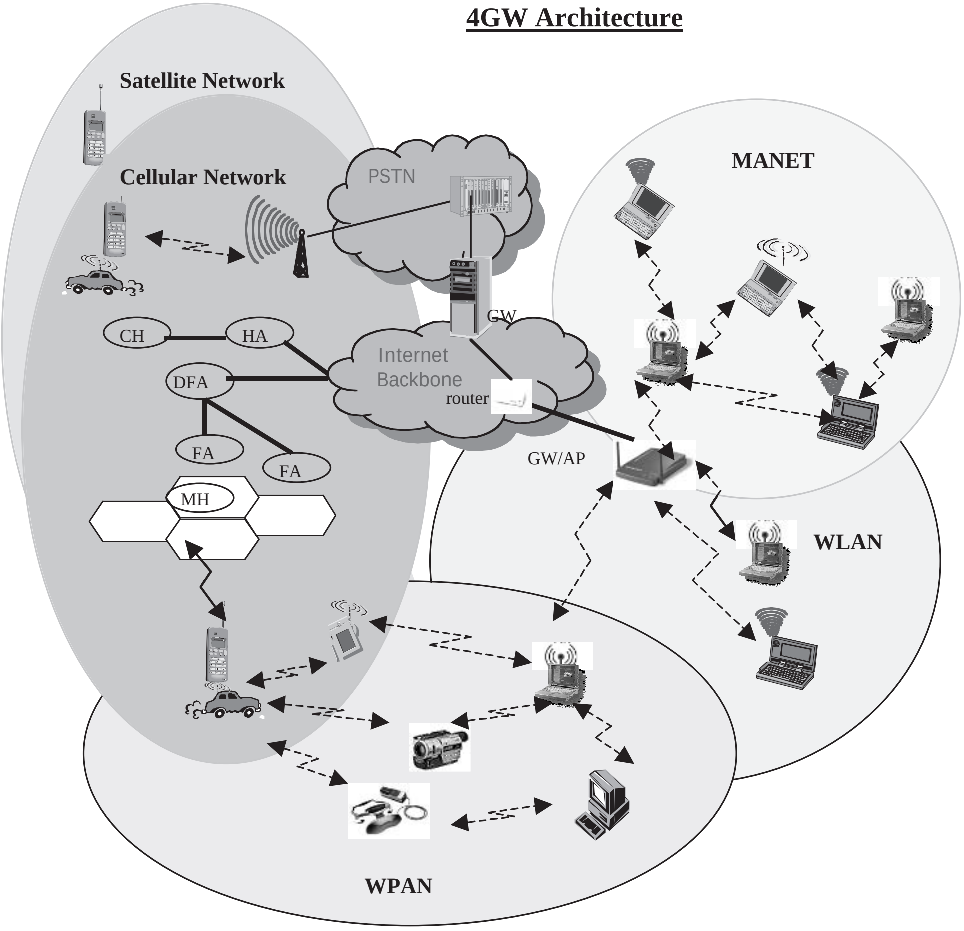


Figure 1.1. 4G wireless network architecture.

1.2.4.1. Network Integration.

4G networks are touted as the hybrid broadband networks that integrate different network topologies and platforms. In Figure 1.1, the integration of various types of networks in 4G is represented by the overlapping of different network boundaries. There are two levels of integration: the first is the integration of heterogeneous wireless networks with varying transmission characteristics such as wireless LAN, WAN, and PAN as well as mobile ad hoc networks; the second level includes the integration of wireless networks and fixed network-backbone infrastructure, the Internet and PSTN.

1.2.4.2. All-IP Networks.

4G starts with the assumption that future networks will be entirely packet-switched using protocols evolved from those in use in today's Internet. An all-IP-based 4G wireless network has intrinsic advantages over its predecessors. IP is compatible with, and independent of, the actual radio access technology. This means that the core 4G network can be designed and can evolve independently from access networks. Using an IP-based core network also means the immediate tapping of the rich protocol suites and services already available, for example, voice and data convergence, can be supported by using a readily available VoIP set of protocols such as MEGACOP, MGCP, SIP, H.323, and SCTP. Finally, the converged all-IP wireless core networks will be packet based and support packetized voice and multimedia on top of data. This evolution is expected to greatly simplify the networks and reduce cost for maintaining separate networks for different traffic types.

1.2.4.3. Lower Cost and Higher Efficiency. 4G IP-based systems are expected to be cheaper and more efficient. First, equipment costs are four to ten times lower than equivalent circuit-switched equipment for 2G and 3G wireless infrastructures. An open converged IP wireless environment further reduces costs for network buildout and maintenance. There will be no need to purchase extra spectrum, as 2G/3G spectrum can be reused in 4G and much of the spectrum needed by WLAN and WPAN is public and does not require a license.

1.2.4.4. Ultrahigh Speed and Multimedia Applications. 4G systems aim to provide ultrahigh transmission speeds of up to 100 Mbps, 50 times faster than those in 3G networks. This leap in transmission speed will enable high-bandwidth wireless services, allowing users to watch TV, listen to music, browse the Internet, access business programs, perform real-time video streaming, and other multimedia-oriented applications, such as E-Commerce, as if they were sitting at home or in the office.

1.2.4.5. Ubiquitous Computing. A major goal toward the 4G Wireless evolution is the provision of pervasive computing environments that can seamlessly and ubiquitously support users in accomplishing their tasks, in accessing information or communicating with other users at any time, anywhere, and from any device. In this environment [172], computers get pushed further into the background; computing power and network connectivity are embedded in virtually every device to bring computation to us, no matter where we are or under what circumstances we work. These devices will personalize themselves in our presence to find the information or software needed.

1.2.4.6. Support of Ad Hoc Networking. Noninfrastructure-based mobile ad hoc networks (MANETs) are expected to become an important part of the 4G architecture. An ad hoc mobile network is a transient network formed dynamically by a collection of arbitrarily located wireless mobile nodes without the use of existing network infrastructure or centralized administration. Mobile ad hoc networks are gaining momentum because they help realize network services for mobile users in areas with no preexisting communications infrastructure [8]. Ad hoc Networking enables independent wireless nodes, each limited in transmission and processing power, to be "chained" together to provide wider networking coverage and processing capabilities. The nodes can also be connected to a fixed-backbone network through a dedicated gateway device, enabling IP networking services in areas where Internet services are not available due to lack of preinstalled infrastructure. All these advantages make ad hoc networking an attractive option in the future wireless networks arena.

1.2.4.7. Location Intelligence. To support ubiquitous computing requirements, 4G terminals need to be more intelligent in terms of user's locations and service needs, including recognizing and being adaptive to user's changing geographical positions, as well as offering location-based services [94]. Anytime, anywhere requires the intelligent use of location information and the embedding of this information in various applications.

Outdoor wireless applications can use the Global Positioning System (GPS) to obtain location information. GPS is a satellite-based system that can provide easy and relatively accurate positioning information almost anywhere on earth. Many GPS implementations are available, including integrating a GPS receiver into a mobile phone (GPS/DGPS), or adding fixed GPS receivers at regular intervals to obtain data to complement readings on a phone (A-GPS), or by using help from fixed base stations (E-OTD). These implementations provide different fix times and accuracies ranging from 50 m to 125 m. For indoor applications, since GPS signals cannot be received well inside buildings, alternative technologies like infrared, ultrasound, or radio have to be used.

Possible location-based services include finding nearest service providers, e.g., restaurants and cinemas; searching for special offers within an area; warning of traffic or weather situations; sending advertisements to a specific area; searching for other collocated users; active badge systems, and so on.

Location information can also be used to help enhance other 4G network services; for example, by using location information to aid and optimize routing in mobile ad hoc networks. Geocasting is another new application that involves broadcasting messages to receivers within a user-defined geographical area.

1.3. MOBILE AD HOC NETWORKS

As mentioned in Section 1.2.4, mobile ad hoc networks (MANETs) are envisioned to become key components in the 4G architecture, and ad hoc networking capabilities are expected to become an important part of overall next-generation wireless network functionalities. In general, mobile ad hoc networks are formed dynamically by an autonomous system of mobile nodes that are connected via wireless links without using an existing network infrastructure or centralized administration. The nodes are free to move randomly and organize themselves arbitrarily; thus, the network's wireless topology may change rapidly and unpredictably. Such a network may operate in a standalone fashion, or may be connected to the larger Internet. Mobile ad hoc networks are infrastructureless networks since they do not require any fixed infrastructure such as a base station for their operation. In general, routes between nodes in an ad hoc network may include multiple hops and, hence, it is appropriate to call such networks "multihop wireless ad hoc networks." Figure 1.2 shows an example mobile ad hoc network and its communication topology.

As shown in Figure 1.2, an ad hoc network might consist of several home-computing devices, including notebooks, handheld PCs, and so on. Each node will be able to communicate directly with other nodes that reside within its transmission range. For communicating with nodes that reside beyond this range, the node needs to use intermediate nodes to relay messages hop by hop.

1.3.1. Characteristics and Advantages

MANETs inherit common characteristics found in wireless networks in general, and add characteristics specific to ad hoc networking:

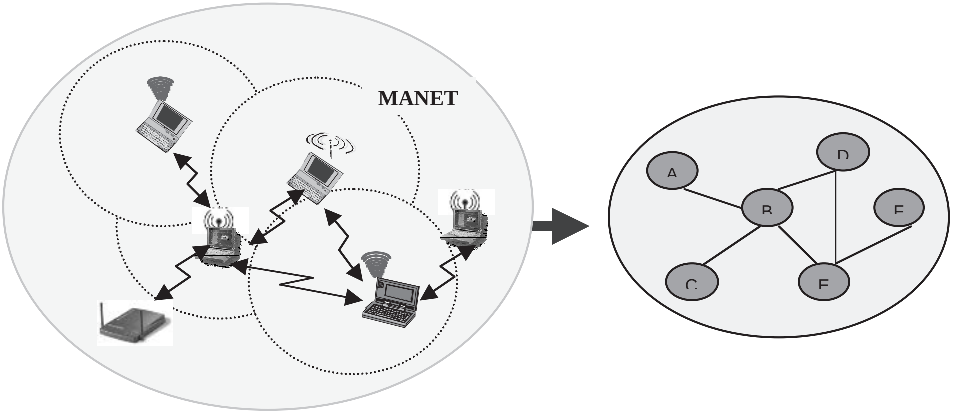


Figure 1.2. Mobile ad hoc network.

- *Wireless.* Nodes communicate wirelessly and share the same media (radio, infrared, etc.).
- *Ad-hoc-based.* A mobile ad hoc network is a temporary network formed dynamically in an arbitrary manner by a collection of nodes as need arises.
- *Autonomous and infrastructureless.* MANET does not depend on any established infrastructure or centralized administration. Each node operates in distributed peer-to-peer mode, acts as an independent router, and generates independent data.
- *Multihop routing.* No dedicated routers are necessary; every node acts as a router and forwards each others' packets to enable information sharing between mobile hosts.
- *Mobility.* Each node is free to move about while communicating with other nodes. The topology of such an ad hoc network is dynamic in nature due to constant movement of the participating nodes, causing the intercommunication patterns among nodes to change continuously.

Ad hoc wireless networks eliminate the constraints of infrastructure and enable devices to create and join networks on the fly—any time, anywhere—for virtually any application.

1.3.2. MANET Applications

Because ad hoc networks are flexible networks that can be set up anywhere at any time, without infrastructure, including preconfiguration or administration, people have come to realize the commercial potential and advantages that mobile ad hoc networking can bring. Next we will look at the range of mobile ad hoc network applications, how they evolved historically, and will evolve in the future.

Historically, mobile ad hoc networks have primarily been used for tactical network-related applications to improve battlefield communications and survivability. The dynamic nature of military operations means it is not possible to rely on access to a fixed preplaced communication infrastructure on the battlefield. Pure wireless communication also has the limitation that radio signals are subject to interference and radio frequencies higher than 100 MHz rarely propagate beyond line of sight (LOS) [16]. A mobile ad hoc network creates a suitable framework to address these issues, provides a mobile wireless distributed multihop wireless network without preplaced infrastructure, and provides connectivity beyond LOS.

Early ad hoc networking applications can be traced back to the DARPA Packet Radio Network (PRNet) project in 1972 [16]. This was primarily inspired by the efficiency of packet switching technology, such as bandwidth sharing and store-and-forward routing, and its possible application in mobile wireless environments. PRNet featured a distributed architecture consisting of networks of broadcast radios with minimal central control, and a combination of Aloha and CSMA channel access protocols used to support the dynamic sharing of the broadcast radio channel. In addition, by using multihop store-and-forward routing techniques, the radio coverage limitation is removed, which effectively enables multiuser communication within a very large geographic area.

Survivable Radio Networks (SURANs) were developed by DARPA in1983 to address open issues in PRNet, in the areas of network scalability, security, processing capability, and energy management. The main objectives of this effort were to develop network algorithms to support networks that can scale to tens of thousands of nodes and withstand security attacks, as well as use small, low-cost, low-power radio that could support sophisticated packet radio protocols [16]. This effort resulted in the design of Low-cost Packet Radio (LPR) technology in 1987 [17], which featured a digitally controlled DS spread-spectrum radio with an integrated Intel 8086 microprocessor-based packet switch. In addition, a family of advanced network management protocols was developed, and hierarchical network topology based on dynamic clustering was used to support network scalability. Other improvements in radio adaptivity, security and increased capacity were achieved through management of spreading keys [18].

Toward the late 1980s and early 1990s, the growth of the Internet infrastructure and the microcomputer revolution created a feasible environment for the implementation of the initial packet radio network ideas [16]. To leverage the global information infrastructure in the mobile wireless environment, the U.S. Department of Defense initiated the DARPA Global Mobile (GloMo) Information Systems program in 1994 [20], which aimed to support Ethernet-type multimedia connectivity any time, anywhere, among wireless devices. Several networking designs were explored; for example, Wireless Internet Gateways (WINGs) at UCSC deploys a flat peer-to-peer network architecture, whereas the Multimedia Mobile Wireless Network (MMWN) project from GTE Internetworking uses a hierarchical network architecture that is based on clustering techniques.

Tactical Internet (TI), implemented by U.S. Army in 1997, is by far the largest-scale implementation of mobile wireless multihop packet radio network [16]. TI uses direct-sequence, spread-spectrum, time division multiple access radio with data rates in the tens of kilobits per second ranges, whereas modified commercial Internet protocols are used for networking among nodes.

Extending the Littoral Battle-space Advanced Concept Technology Demonstration (ELB ACTD) in 1999 is another MANET deployment exploration to demonstrate the feasibility of Marine Corps war fighting concepts that require over-the-horizon (OTH) communications from ships at sea to Marines on land via an aerial relay. Approximately two dozen nodes were configured for the network, Lucent's WaveLAN and VRC-99A were used to build the access and backbone network connections. The ELB ACTD was successful in demonstrating the use of aerial relays for connecting users beyond LOS.

Although early MANET applications and deployments were military oriented, nonmilitary applications have grown substantially since then and have become the main focus today. Especially in the last few years, with the rapid advances in mobile ad hoc networking research, mobile ad hoc networks have attracted considerable attention and interest from the commercial sector as well as the standards community. The introduction of new technologies such as Bluetooth, IEEE 802.11, and Hyperlan greatly facilitate the deployment of ad hoc technology outside of the military domain. As a result, many new ad hoc networking applications have since been conceived to help enable new commercial and personal communications beyond the tactical networks domain, including personal area networking, home networking, law enforcement operations, search-and-rescue operations, commercial and educational applications, sensor networks, and so on. Table 1.2 shows the classification of present and future applications as well as the example services they provide.

1.3.3. Design Issues and Constraints

As described in the previous section, the ad hoc architecture has many benefits, such as self-reconfiguration, ease of deployment, and so on. However, this flexibility and convenience come at a price. Ad hoc wireless networks inherit the traditional problems of wireless communications, such as bandwidth optimization, power control, and transmission quality enhancement [8], while, in addition, their mobility, multihop nature, and the lack of fixed infrastructure create a number of complexities and design constraints that are new to mobile ad hoc networks, as discussed in the following subsections.

1.3.3.1. They are Infrastructureless. Mobile ad hoc networks are multihop infrastructureless wireless networks. This lack of fixed infrastructure in addition to being wireless, generate new design issues compared with fixed networks. Also, lack of a centralized entity means network management has to be distributed across different nodes, which brings added difficulty in fault detection and management.

1.3.3.2. Dynamically Changing Network Topologies. In mobile ad hoc networks, since nodes can move arbitrarily, the network topology, which is typically multihop, can change frequently and unpredictably, resulting in route changes, frequent network partitions, and, possibly, packet losses [12, 36].

1.3.3.3. Physical Layer Limitation. The radio interface at each node uses broadcasting for transmitting traffic and usually has limited wireless transmission range, resulting in specific mobile ad hoc network problems like hidden terminal problems, exposed terminal problem, and so on. Collisions are inherent to the medium, and there is a higher probability of packet losses due to transmission errors compared to wireline systems.

1.3.3.4. Limited Link Bandwidth and Quality. Because mobile nodes communicate with each other via bandwidth-constrained, variable capacity, error-prone, and insecure wireless channels, wireless links will continue to have significantly lower capacity than wired links and, hence, congestion is more problematic.

1.3.3.5. Variation in Link and Node Capabilities. Each node may be equipped with one or more radio interfaces that have varying transmission/receiving capabilities

Table 1.2. Mobile Ad hoc Network Applications

Applications	Descriptions/Services
Tactical networks	Military communication, operations Automated Battlefields
Sensor networks [25]	Collection of embedded sensor devices used to collect real-time data to automate everyday functions. Data highly correlated in time and space, e.g., remote sensors for weather, earth activities; sensors for manufacturing equipment.
	Can have between 1000–100,000 nodes, each node collecting sample data, then forwarding data to centralized host for processing using low homogeneous rates.
Emergency services	Search-and-rescue operations as well as disaster recovery; e.g., early retrieval and transmission of patient data (record, status, diagnosis) from/to the hospital.
	Replacement of a fixed infrastructure in case of earthquakes, hurricanes, fire, etc.
Commercial environments	E-Commerce, e.g., electronic payments from anywhere (i.e., in a taxi).
	Business: dynamic access to customer files stored in a central location on the fly provide consistent databases for all agents mobile office
	Vehicular Services: transmission of news, road conditions, weather, music local ad hoc network with nearby vehicles for road/accident guidance
Home and enterprise networking	Home/office wireless networking (WLAN), e.g., shared whiteboard application, use PDA to print anywhere, trade shows
	Personal area network (PAN)
Educational applications	Set up virtual classrooms or conference rooms
	Set up ad hoc communication during conferences, meetings, or lectures
Entertainment	Multiuser games Robotic pets Outdoor Internet access
Location-aware services	Follow-on services, e.g., automatic call forwarding, transmission of the actual workspace to the current location
	Information services push, e.g., advertise location-specific service, like gas stations pull, e.g., location-dependent travel guide; services (printer, fax, phone, server, gas stations) availability information; caches, intermediate results, state information, etc.

and operate across different frequency bands [130, 137]. This heterogeneity in node radio capabilities can result in possibly asymmetric links. In addition, each mobile node might have a different software/hardware configuration, resulting in variability in processing capabilities. Designing network protocols and algorithms for this heterogeneous network can be complex, requiring dynamic adaptation to the changing power and channel conditions, traffic load/distribution variations, load balancing, congestion, and service environments.

1.3.3.6. Energy Constrained Operation. Because batteries carried by each mobile node have limited power, processing power is limited, which in turn limits services and applications that can be supported by each node. This becomes a bigger issue in mobile ad hoc networks because as each node is acting as both an end system and a router at the same time, additional energy is required to forward packets from other nodes [23].

1.3.3.7. Network Robustness and Reliability. In MANET, network connectivity is obtained by routing and forwarding among multiple nodes. Although this replaces the constraints of fixed infrastructure connectivity, it also brings design challenges. Due to various conditions like overload, acting selfishly, or having broken links, a node may fail to forward the packet. Misbehaving nodes and unreliable links can have a severe impact on overall network performance. Lack of centralized monitoring and management points means these types of misbehaviors cannot be detected and isolated quickly and easily, adding significant complexity to protocol design.

1.3.3.8. Network Security. Mobile wireless networks are generally more vulnerable to information and physical security threats than fixed-wireline networks. The use of open and shared broadcast wireless channels means nodes with inadequate physical protection are prone to security threats. In addition, because a mobile ad hoc network is a distributed infrastructureless network, it mainly relies on individual security solution from each mobile node, as centralized security control is hard to implement. Some key security requirements in ad hoc networking include:

- Confidentiality: preventing passive eavesdropping
- Access control: protecting access to wireless network infrastructure
- Data integrity: preventing tampering with traffic (i.e., accessing, modifying or injecting traffic)
- Denial of service attacks by malicious nodes

1.3.3.9. Network Scalability. Current popular network management algorithms were mostly designed to work on fixed or relatively small wireless networks. Many mobile ad hoc network applications involve large networks with tens of thousands of nodes, as found, for example, in sensor networks and tactical networks [16]. Scalability is critical to the successful deployment of such networks. The evolution toward a large network consisting of nodes with limited resources is not straightforward and presents many challenges that are still to be solved in areas such as addressing, routing, location management, configuration management, interoperability, security, high-capacity wireless technologies, and so on.

1.3.3.10. Quality of Service. A quality of service (QoS) guarantee is essential for successful delivery of multimedia network traffic. QoS requirements typically refer to a wide set of metrics including throughput, packet loss, delay, jitter, error rate, and so on [150]. Wireless and mobile ad hoc specific network characteristics and constraints described above, such as dynamically changing network topologies, limited link bandwidth and quality, variation in link and node capabilities, pose extra difficulty in achieving the required QoS guarantee in a mobile ad hoc network.

1.4. TECHNICAL CHALLENGES AND RESEARCH OVERVIEW

The specific MANET issues and constraints described in the previous section present a host of challenges in ad hoc network design. A significant body of research has been accumulated to address these specific issues and constraints. In this section, we describe some of the main research areas within the mobile ad hoc network domain. Figure 1.3 shows the MANET network layers and the corresponding research issues associated with each layer.

1.4.1. Media Access Control and Optimization

In MANET, use of broadcasting and shared transmission media introduces a nonnegligible probability of packet collisions and media contention. In addition, with half-duplex radio, collision detection is not possible, which severely reduces channel utilization as well

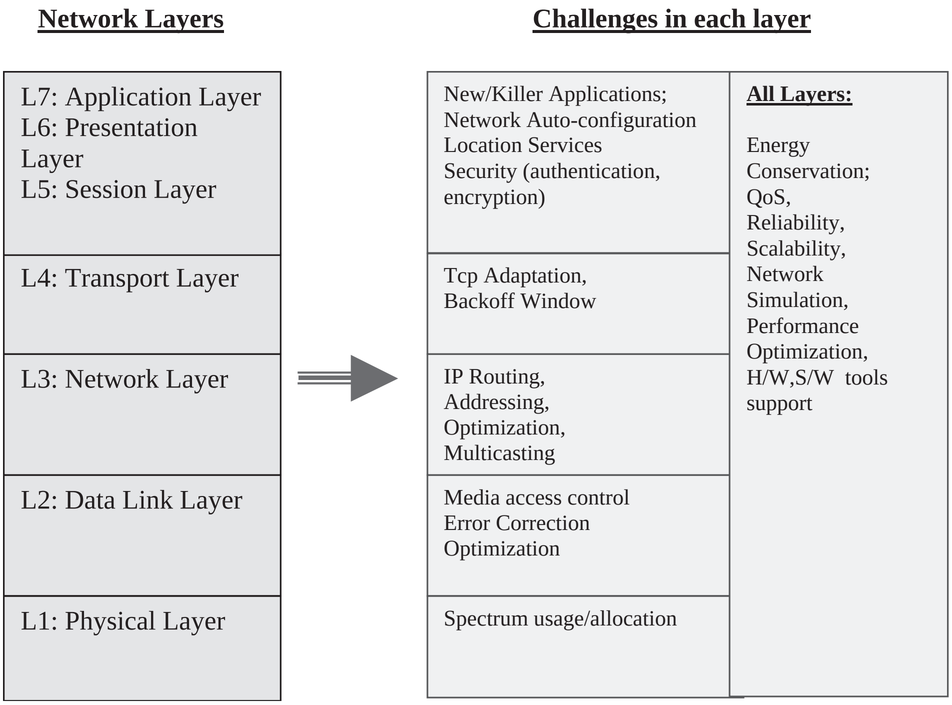


Figure 1.3. MANET network layers and research challenges.

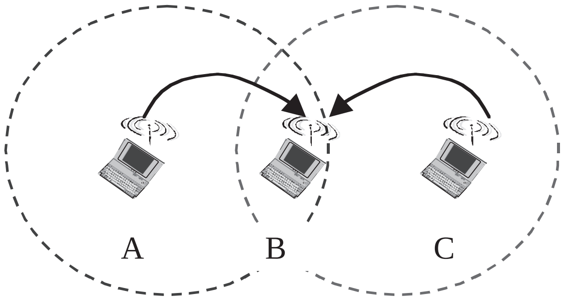


Figure 1.4. Hidden-terminal problem.

as throughput, and brings new challenges to conventional CSMA/CD-based and MAC protocols in general. Among the top issues are the hidden-terminal and exposed-terminal problems.

The hidden-terminal problem occurs when two (or more) terminals, say, A and C, cannot detect each other's transmissions (due to being outside of each other transmission range) but their transmission ranges are not disjoint [38, 152]. As shown in Figure 1.4, a collision may occur, for example, when terminal A and C start transmitting toward the same receiver, terminal B in the figure.

The exposed-terminal problem results from situations in which a permissible transmission from a mobile station (sender) to another station has to be delayed due to the irrelevant transmission activity between two other mobile stations within sender's transmission range.

Figure 1.5 depicts a typical scenario in which the exposed-terminal problem may occur. Let us assume that terminals A and C can hear transmissions from B, but terminal A cannot hear transmissions from C. Let us also assume that terminal B is transmitting to terminal A, and terminal C has a frame to be transmitted to D. According to the CSMA scheme, C senses the medium and finds it busy because of B's transmission, and, therefore, refrains from transmitting to D, although this transmission would not cause a collision at A. The exposed-terminal problem may thus result in loss of throughput.

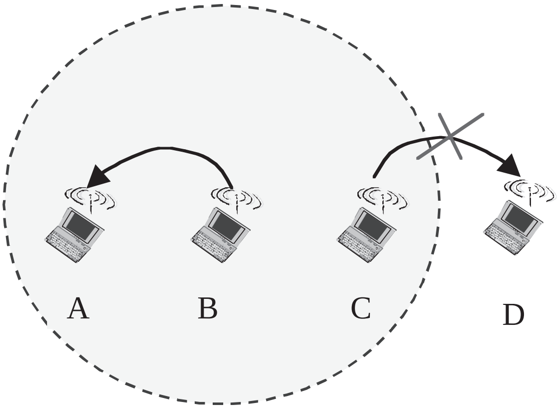


Figure 1.5. Exposed-terminal problem.

The very early access protocols such as Aloha, CSMA, Bram and, TDMA introduced in the 1970s [7] were primarily intended as solutions to multiaccess channels, such as any broadcast media, similar to early LANs, and quickly proved inadequate to effectively deal with the needs of current-day ad hoc network applications. The first protocols designed specifically for mobile and multihop mobile networks [37, 130, 137] were designed with tactical communication in mind and were based on slotted channels requiring rigid synchronization. As recent ad hoc technologies started to take shape, a very large number of new-generation ad hoc protocols such as MACA (multiple access with collision avoidance protocol), MACAW (MACA with CW optimization), FAMA (floor acquisition multiple access), MACA/PR and MACA-BI (multiple access with collision avoidance by invitation protocol) [39–44] have been proposed to resolve the various hidden-terminal, exposed-terminal and similar problems, and improve channel performance in MANET. The key ideas behind these protocols involve sending RTS (request to send) and CTS (clear to send) packets before the data transmission has actually taken place [38]. When a node wishes to transmit a packet to a neighbor, it first transmits a RTS packet. The receiver then consents to the communication by replying with a CTS packet. On hearing the CTS, the sender can transmit its data packet.

For example, a virtual carrier sensing mechanism based on the RTS/CTS mechanism has been included in the 802.11 standard to alleviate the hidden-terminal problem that may occur by using physical carrier sensing only. Virtual carrier sensing is achieved by using two control frames, Request To Send (RTS) and Clear To Send (CTS), before the data transmission actually takes place. Specifically, before transmitting a data frame, the source station sends a short control frame, named RTS, to the receiving station, announcing the upcoming frame transmission. Upon receiving the RTS frame, the destination station replies by a CTS frame to indicate that it is ready to receive the data frame. Both the RTS and CTS frames contain the total duration of the transmission, that is, the overall time interval needed to transmit the data frame and the related ACK. This information can be read by any station within the transmission range of either the source or the destination station. Hence, stations become aware of transmissions from hidden stations, and the length of time the channel will be used for these transmissions.

However, studies [45, 46] show that when traffic is heavy, a data packet can still experience collision due to loss/collision of RTS or CTS packets. To alleviate this problem, comprehensive collision-avoidance mechanisms have been introduced via a backoff mechanism. In principle, once a transmitting node senses an idle channel, it waits for a random backoff duration (determined by a contention window, and increasing exponentially with each reattempt) before attempting to transmit the packet, and congestion control is achieved by dynamically choosing the contention window based on the traffic congestion situation in the network. Besides backoff methods, other mechanisms have also been proposed to address this problem. DBTMA (dual busy tone multiple access) [46] provides a scheme whereby special signals called busy tones (BTt/BTr) are used to prevent other mobile hosts unaware of the earlier RTS/CTS dialogues from destroying the ongoing transmission. The distributed collision resolution protocol EMMCRR [50] uses power control and energy measurement techniques to achieve efficient collision avoidance; and in [38], a combination of RTS/CTS, power control and busy-tone techniques are used to further increase channel utilization.

In IEEE 802.11, CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance), a variation of the MACA protocol, is used for the MAC layer, and DCF is used to provide collision avoidance and congestion control [47].

Besides collision avoidance, other optimization studies have been done in the MAC layer to improve MANET performance, including MAC improvement and algorithms used to reduce mobile node energy consumption [26, 38] as well as MAC optimizations for improving TCP layer performance.

1.4.2. Ad Hoc Routing

The highly dynamic nature of mobile ad hoc networks results in frequent changes and unpredictability in network topologies, adding difficulty and complexity to routing among the mobile nodes within the network. These added challenges, coupled with the critical importance of routing protocols in establishing communications among mobile nodes, make the routing area perhaps the most active research area within the MANET domain. Especially over the last few years, numerous routing protocols and algorithms have been proposed and their performance under various network environments and traffic conditions closely studied and compared. The ultimate goal of the MANET community is to provide a set of standardized protocols that can be both robust and scalable to tens of thousands of network nodes to enable fast commercialization of mobile ad hoc networks in increasing network applications suites.

The primary objective of an ad-hoc network routing protocol is the correct and efficient route establishment between a pair of nodes so that messages may be delivered reliably and in a timely manner. Route construction should be done with minimum overhead and bandwidth consumption [57]. Existing distance-vector and link-state-based routing protocols are designed for static environment, and are, therefore, unable to catch up with frequent topology changes of ad hoc environments, resulting in degradation in performance, including slow route convergence, low communication throughput [23], possible route loops during node failure, and network partition or congestion. In addition, protocols that use flooding techniques, such as link-state algorithms, tend to create excessive traffic and control overhead during route establishment. New routing protocols need to be designed to suit the specific needs of mobile ad hoc network environments and characteristics, particularly mobility and bandwidth/energy limitations.

Important criteria and considerations used in designing and comparing the new routing protocols include:

- Simplicity and ease of implementation
- Rapid route convergence—routes should be loop-free and optimal, and, possibly, multiple routes should be available between each pair of nodes to increase robustness
- Distributed but lightweight in nature—can quickly adapt to changes in topology and traffic pattern resulting from mobility and failure conditions; protocol reaction should result in minimal control overhead
- Bandwidth, power, and computing efficient with minimum overhead
- Scalable
- Secure and reliable
- Supporting Quality of Service requirements

In general, ad hoc network routing protocols may be divided into two broad categories: proactive routing protocols and reactive on-demand routing protocols [57]. Proactive routing protocols attempt to maintain consistent, up-to-date routing information between every pair of nodes in the network by propagating, proactively, route updates at fixed time intervals. As the resulting routing information is usually maintained in tables, the protocols are sometimes refered to as table-driven protocols. Reactive on-demand routing protocols, on the other hand, establish a route to a destination only when there is a demand for it, usually initiated by the source node through route discovery process within the network. Once a route has been established, it is maintained by the node until either the destination becomes inaccessible along every path from the source or until the route is no longer used or has expired [57].

Representative proactive protocols include Destination-Sequenced Distance-Vector (DSDV) protocol, Clusterhead Gateway Switch Routing (CGSR) protocol, Wireless Routing Protocol (WRP), Global State Routing (GSR) [79], Optimized Link State Routing Protocol (OLSR), Fisheye State Routing (FSR) [75] Protocol, Landmark Routing (LANMAR) Protocol, and Hierarchical State Routing (HSR).

The Destination-Sequenced Distance-Vector (DSDV) protocol (PB94) is a distance vector protocol with extensions to make it suitable for MANET. Every node maintains a routing table with one route entry for each destination in which the shortest path route (based on number of hops) is recorded. To avoid routing loops, a destination sequence number is used. A node increments its sequence number whenever a change occurs in its neighborhood. This number is used to select among alternative routes for the same destination. Nodes always select the route with the greatest sequence number.

CGSR extends DSDV with a cluster framework concept that increases protocol scalability [63]; also, heuristic methods like priority token scheduling, gateway code scheduling, and path reservation [63] are used to improve the protocol's performance. On the other hand, setting up structure in a highly dynamic environment can adversely affect protocol performance since the structure might not persist for a very long time.

WRP is another loop-free proactive protocol whereby four tables are used to maintain distance, link cost, routes, and message retransmission information [64]. General route updates are sent among neighboring nodes with distance and second-to-last hop information for each destination, resulting in faster convergence.

Despite the variance in number of routing tables used and the difference in routing information maintained in these tables, proactive routing protocols like DSDV, CGSR, and WRP are all distance vector shortest-path-based and have the same degree of complexity during link failures and additions.

The OLSR protocol [74] is an optimization from the pure Link State protocol. In OLSR, instead of flooding a node's complete link state information to all nodes in the network, only link information from a subset of links to the neighboring Multipoint Relay Selectors (MRS) are flooded to other MRSs, which then relay the control information to their neighbor nodes. Upon receiving the updates, each node calculates the routes to all known nodes, which are a sequence of hops through multipoint relay nodes to each destination node. All nodes select their set of multipoint relays such that the set covers all the nodes that are two hops away. The FSR protocol [75, 76] is also an optimization over Link State algorithm using the fisheye technique. In essence, FSR will propagate link state information to other nodes in the network based on how far away (defined by scopes, which are determined by number of hops) the nodes are. The protocol will propagate link state information more frequently with nodes that are in a closer scope as opposed to ones that are further away. This means that a route will be less accurate the further away the node is, but once the message gets closer to the destination, the accuracy increases. LANMAR

[77, 78] builds on top of FSR and achieves hierarchical routing by partitioning the network nodes into different mobility groups. A landmark node is elected within each group to keep track of which logical subnet a node belongs to and facilitate intergroup routing, whereas FSR is used for intragroup routing. OLSR and FSR are similar in that they are both Link-State-algorithm-based, and both optimizations provide the same benefit of reducing routing overhead and, consequently, improve bandwidth efficiency.

Representative reactive routing protocols include Dynamic Source Routing (DSR), Ad-hoc on-Demand Distance Vector (AODV), Temporally Ordered Routing Algorithm (TORA), Associativity-Based Routing (ABR), and Signal Stability Routing (SSR). DSR is a loop-free source based on the demand routing protocol [65], in which each node maintains route caches that contain the source routes learned by the node, and the route discovery process is only initiated when a source node does not already have a valid route to the destination in its route cache. Entries in the route cache are continually updated as new routes are learned. AODV is an improvement on the DSDV protocol. AODV minimizes the number of route broadcasts by creating routes on an on-demand basis [61], as opposed to maintaining a complete list of routes as in the DSDV algorithm. Just like DSR, route discovery is initiated on an on-demand basis, the route request is then forward to the neighbors, and so on, until either the destination or an intermediate node with a fresh route to the destination are located. DSR has potentially larger control overhead and memory requirements than AODV since each DSR packet must carry full routing path information, whereas in AODV packets only contain the destination address. On the other hand, DSR can utilize both asymmetric and symmetric links during routing, whereas AODV only works with symmetric links, a condition that is more difficult to satisfy in mobile wireless environments. In addition, nodes in DSR maintain multiple routes to a destination in the cache, which is helpful during link failure. In general, both AODV and DSR work well in small-to-medium-sized networks with moderate mobility.

TORA is another source-initiated on-demand routing protocol, built on the concept of link reversal [58] of Directed Acyclic Graph (ACG). In addition to being loop-free and bandwidth-efficient, TORA has the property of being highly adaptive and quick in route repair during link failure, while providing multiple routes for any desired source/destination pair. These features make it especially suitable for large highly dynamic mobile ad hoc environments with dense populations of nodes. The limitation in TORA's applicability comes from its reliance on synchronized clocks. If a node does not have a GPS positioning system or some other external time source, or if the time source fails, the algorithm cannot be used.

The ABR protocol is also a loop-free protocol using a new routing metric termed "degree of association stability" in selecting routes [66] so that routes discovered can be longer-lived routes and thus be more stable and require less updating in the future. The limitation of ABR mainly comes from the periodic beaconing requirement used to establish the association stability metrics, which may result in additional energy consumption. The Signal Stability Algorithm (SSA) [84] is basically an ABR with the additional property of selecting routes based on signal strength of the link.

In general, on-demand reactive protocols are more efficient than proactive routing protocols in terms of control overhead and power consumption since routes are only established when required. By contrast, proactive protocols require periodic route updates to keep information current and consistent. In addition, many routes maintained might never be needed, which significantly adds to routing overhead in the bandwidth-constrained network. As routing overhead grows exponentially with network size, it prevents the application of these protocols in large-scaled networks.

Proactive routing protocols generally provide better quality of service than on-demand protocols. As in proactive protocols, routing information is constantly updated, routes to every destination are always available and up-to-date, and, hence, end-to-end delay can be minimized. For on-demand protocols, the source node has to wait for the route to be discovered before communication can happen. This latency in route discovery might be intolerable for real-time communications. In [57], the authors present a set of tables that summarize the main differences between these protocols in terms of their complexity, route update patterns, and capabilities.

The above categorization is very broad, and other taxonomies exist to categorize routing protocols in ad hoc domains based on different selection criteria, including those based on structure, type of cast, cost functions (metrics) [86], and so on. For example, some protocols rely on setting up an internal structure, hierarchy, or partitions to help improve routing efficiency and achieve scalability. CGSR and HSR use a cluster/cluster-head-based hierarchical architecture, OLSR protocol relies on the selection of multipoint relays, FSR partitions nodes based on scope, the LANMAR method depends on landmark node and hierarchical partitions of the network nodes, CEDAR relies on core nodes to negotiate routes and maintain and disseminate QoS information, and the Zone-Based Hierarchical Link State Routing Protocol (ZRP) [80] divides the network into nonoverlapping zones based on nodes' geolocation, with neighboring zone connectivity information propagated by dedicated gateway nodes. In general, structure-based protocols are more bandwidth-efficient and scalable, but the cost to maintain the structure can be prohibitive, especially in mobile ad hoc environments where constant topology changes might result in unstable structures.

Yet another category of routing protocols base their routing decisions on the nodes' geographical position, such as the one provided by GPS [55] or other mechanisms [184, 185]. These are typically referred as location-aware routing or position-based routing algorithms. Location-aware routing does not require the routes' establishment and maintenance. No routing information is stored. Typically, a node selects the next hop for packets' forwarding by using the physical position of its one-hop neighbors and the physical position of the destination node; positioning information of the networks' nodes are usually obtained via queries offered through some location service. The packets are then forwarded to a neighbor in the receiver's direction. The use of geolocation information avoids network-wide searches, as both control and data packets are sent toward the known geographical coordinates of the destination node. These features make location-aware routing protocols quickly adaptive to route changes, and more scalable than unicast protocols such as AODV, DSDV, or DSR. DREAM, the first location-based routing protocol, utilized the idea of the "distance factor," allowing optimized performance to be obtained based on the observation that the further the nodes are from each other, the slower they appear to be moving [87]. Hence, less routing information has to be transferred between remote nodes. As demonstrated in [87], location-based protocols can yield an order of magnitude improvement in performance and bandwidth/energy usage by utilizing location information in routing.

A large number of location-aware algorithms have been proposed in the literature since then. As represented by [186–189], these protocols typically fall into three main strategies: greedy forwarding, directed flooding, and hierarchical routing. Greedy forwarding and directed flooding algorithms forward the packet to one or more neighbors, respectively. Hierarchical routing algorithms are a combination of position-based, and non-position-based routing algorithms. Proposed greedy forwarding algorithms include The Most Forward within Radius policy (MFR) [190], the Nearest with Forward Progress scheme (NFP) [191] and the compass routing scheme [192]; representative directed flooding algorithms include DREAM [87] and LAR [193]; and representative hierarchical routing algorithms include the Grid Routing [194] and the Terminode Routing protocols [195].

In [86], the authors present a detailed discussion of applications of different routing protocols in various types of networks. In general, they show that the proactive protocols are sufficient for a small-scale static network, whereas reactive protocols such as DSR and AODV normally work well for medium-size networks with moderate mobility. For large-scale networks with hundreds or thousands of nodes, in which layering and partitioning are essential in ensuring network performance, structure-based or hybrid protocols are more appropriate. Overall, a good approach in routing protocol design might come from hybrid routing protocols; for example, by using proactive protocols in local zones, while using reactive protocols between zones (ZRP), by using location-aware routing for nodes over long distances when the forwarding node and the receiver are far away, or by incorporating a hierarchical clustering algorithm to AODV to increase scalability. Location-based algorithms may also become key contenders for solving efficient routing in MANET networks as location based services become generally available.

1.4.3. Multicasting and Broadcasting

As mentioned in Section 1.4.2, routing protocols can also be classified by the type of cast property, that is, whether they use Unicast, Geocast, Multicast, or Broadcast [86]. The first works on broadcasting in multihop and mobile multihop wireless ad hoc networks focused on slotted channels [90, 92]. Broadcast is a basic mode of operation in the wireless medium whereby messages are sent to all source nodes' neighbors. In a unicast operation, a single source transmits messages or data packets to one destination. Unicast protocols serve as the basis for other types of protocols. Multicast routing protocols come into play when a node needs to send the same message or stream of data to multiple destinations, whereas Geocast protocols are used to deliver data packets to groups of nodes situated in a specified geographical area. Geocast is a special type of Multicast. In a multicast situation, nodes may join or leave a multicast group as desired, and in Geocast, nodes can only join or leave the group by entering or leaving the defined geographical region [88]. To operate properly, Geocast typically requires additional node location information from positioning systems, such as GPS.

Multicasting is an efficient communication service for supporting multipoint applications (e.g., software distributions, audio/video conferencing) in the Internet. In MANET, the role of multicast services is potentially even more important due the bandwidth and energy savings that can be achieved through multicast packets' delivery [CW87]. Conventional static multicast routing protocols work well under static configurations. Supporting multicast routes under highly dynamic network situation is a major challenge for mobile ad hoc multicasting routing protocols [92].

In general, two mechanisms are used to establish routing information among multicast members within a group: routing tree or mesh. Establishing a routing tree among a group of routers works well in static networks. Multicast meshes are more suitable for dynamic environments as they support higher connectivity than trees. Multicast meshes are established first by flooding control packets within the network. Although multicast meshes perform better than multicast trees in dynamic networks, the mesh mechanism is more inclined to form routing loops. In addition, approaches to mesh building based on flooding incur excessive overhead in large networks [88].

Representative route-tree-based multicast protocols include the Distance Vector Multicast Routing Protocol (DVMRP), AMROUTE, and AODV, DVMRP proactively builds a source-based tree by first flooding the whole network with the multicast traffic and then conducting pruning operations. AODV uses a multicast group leader to establish a core-based tree structure based on demand.

Representative mesh-based multicast routing protocols include the Core-Assisted Mesh Protocol (CAMP), the Forwarding Group Multicast Protocol (FGMP), and the On-Demand Multicast Routing Protocol (ODMRP). These protocols build routing meshes rather than routing trees to disseminate multicast packets within groups. Both FGMP and ODMRP use flooding to build the mesh, whereas CAMP uses one or more core nodes to assist in building the mesh instead of using flooding. In FGMP, the receivers initiate the flooding, whereas the senders initiate the flooding in ODMRP.

To avoid the significant delays in route recovery caused by link failures, in [197] the authors explore the possibility of using a set of precalculated alternate trees. When a link breaks, another tree that does not includes the failed link is deployed. An alternative approach to avoiding problems related to tree/mesh maintenance is implemented in the Explicit Multicasting protocol [198]. This protocol is designed to operate in a stateless manner where no intermediate node needs to maintain multicast forwarding paths.

1.4.4. TCP Issues

TCP was originally designed to work in fixed networks. TCP provides an effective connection-oriented transport control protocol that provides the essential flow control and congestion control required to ensure reliable packet delivery. Because error rates in wired network are quite low, TCP uses packet loss as an indication of network congestion, and deals with this effectively by making corresponding transmission adjustment to its congestion window. The mobile multihop ad hoc environment brings fresh challenges to TCP protocol due to its frequent change in network topology, disconnections, variation in link capability, and high error rate. In a wireless mobile ad hoc network, packet losses are usually not caused by network congestion, but by the high error rate from wireless medium and frequent disconnections from mobility, resulting in backoff mechanisms being inappropriately invoked [36, 99, 101, 103], thus reducing network bandwidth utilization and increasing the delay for connection restoration [102]. In addition, variation in link capability could cause asymmetric links and delayed acknowlegment, which can affect congestion window adjustment as well [98–100]. As a result, standard TCP flow control and congestion control mechanisms do not work well in mobile ad hoc networks.

Besides physical layers issues, a number of studies have shown that MAC layer and network layer protocols can have a significant impact on TCP performance as well. Since link-level reliability is provided by the MAC layer, the error control mechanism used by the MAC layer can adversely affect TCP performance. For example, interaction between TCP and MAC layer backoff timers can cause severe unfairness and capture conditions when CSMA and FAMA are used as MAC layer protocols. Well-defined synchronization is therefore required between the TCP and MAC layer protocols to reduce the effect of this interference on TCP Performance [98, 104].

The multihop routing nature of MANET also contributes, to a certain degree, to loss of performance. Measurements using 2 Mbps 802.11 MAC have shown that TCP throughput decreases by 50% when the traffic moves from the one-hop to the two-hop path [102]. The study in [36, 102] further shows that when the number of hops is small, the throughput decreased with increased number of hops, and was stabilized by effective pipelining only when the number of hops became large enough.

Finally, different TCP implementations can result in different TCP performance; for example, conflicts between TCP data packets and TCP ACKs can cause TCP performance to degrade when window size is greater than 1 packet [104, 105]. Consequently, in order for the TCP protocol to work properly and effectively in MANET, ad hoc specific adaptations are required at various layers. Numerous enhancements and optimizations have been proposed over the past few years to improve TCP performance, many of them developed specifically for wireless cellular networking environments [108–114] where the last hop is based on a wireless medium. Although there are a number of differences between cellular and ad hoc networks, many of these proposed solutions can be readily used in the mobile ad hoc networks [98], whereas others can be used after some adaptation. For example, as packet loss usually results from limitation of the wireless medium, the solution can be to simply retransmit the lost packet to avoid invocation of congestion control mechanisms.

Besides these techniques, numerous new TCP optimization mechanisms have been proposed with the aim of resolving MANET-specific issues, including the adaptation of TCP error detection and recovery strategies in ad hoc environments. For instance, methods have been developed to distinguish between packet losses caused by network congestion/overloading and other factors, such as buffer overflow or transmission errors, and mobility by using link contention information [106], which would allow TCP to take the appropriate action. Techniques have been proposed to minimize the impact of mobility and link disconnection on TCP performance, such as the use of explicit link failure notification (ELFN) [102], a technique to detect and respond to out-of-order packet delivery events [103], as well as link-layer adaptive spacing and link RED methods to adapt TCP for multihop random early detection (RED), like graceful drop behavior [106]. To reduce the interference of the MAC layer on TCP Performance, a new MAC protocol, MACAW, has been proposed to extend MACA by adding link level ACKs and using a less aggressive backoff policy [104]. A combination of link-level protection, backoff policy, and selective queue scheduling techniques has been shown critical for efficient and fair operation of ad hoc networks under TCP [104]. Studies have also been done on comparing different TCP implementations using metrics such as Throughput, Goodput, TransferTime, Weighted Route Length, Average Packet Delay [101], and Expected Throughput [102], as well as on effective TCP implementation techniques, such as how to achieve optimal value for TCP congestion window size to maximize TCP throughput and reduce packet loss [106].

1.4.5. Energy Conservation

Mobile devices rely on batteries for energy. Battery power is finite and represents one of the greater constraints in designing algorithms for mobile devices [153–155]. It is therefore vital that power utilization be managed efficiently by identifying ways to use less power, preferably with no impact on the applications. Energy conservation is not restricted to a single network layer, but instead requires a coordinated effort from all related layers, including the physical-layer transmissions, the operating system, and the applications [156]. Research in this area has focused on several aspects, including study of energy consumption behavior at the network interface level [24] in portable wireless devices, and comparisons of different MAC and routing protocols in terms of their energy conservation capabilities [26, 27]. A first result of designing MAC protocols specifically oriented to reduce energy consumption was given in [34]. A treatment of ARQ issues for wireless channels with the objective of energy minimization was introduced in [196]. Since then, many other proposals for new energy-aware protocols [30, 32, 33, 35] and energy management models/techniques [28, 29, 31] have been made.

A sample study investigating the impact of network technologies on power consumption has been provided in [144]. It has been found that the wireless interface consumes nearly the same amount of energy in the receive, transmit, and idle states, whereas in the sleep state, an interface cannot transmit or receive, and its power consumption is highly reduced. But merely maximizing the time the interface is in power-saving mode (sleep state) is not a viable approach in an ad hoc network environment, as ad hoc networks rely on cooperative efforts among participating nodes to deliver the network service. A greedy node that remains most of the time in a sleep state, without contributing to routing and forwarding, will maximize its battery lifetime but compromise the lifetime of the network.

Strategies have been developed to overcome this problem so that the network interface can be put in a power-saving mode with a minimum impact on transmit and receive operations. These policies typically operate at the physical and MAC layers. For example, at the physical layer, some authors have proposed and analyzed policies (based on monitoring the transmission error rates), that avoid useless transmissions when the channel noise reduces the probability of a successful transmission [157, 158]. At the MAC layer, energy conservation can be achieved by reducing the energy required to successfully transmit a packet, for example by avoiding transmitting when the channel is congested, by synchronizing the node communication time for a single-hop ad hoc network (in 802.11) [161–163], or by finding intervals during which the network interface does not need to be listening [160]. For example, while a node transmits a packet, the other nodes within the same interference and carrier sensing range must remain silent. Therefore, these nodes can sleep with little or no impact on system behavior.

Other strategies have been developed to achieve energy conservation at the overall network level in addition to the node level strategies mentioned above. For example, when a region is dense in terms of nodes, only a small number of them need to be turned on in order to forward the traffic so that the overall network lifetime is optimized.

Controlling the power of the transmitting node is the other main method for achieving power saving in ad hoc networks. Reduced transmission power also allows spatial reuse of frequencies, which can help increase the total throughput of network and minimize multiuser interference [86]. In addition, the probability of intercept and detection is lower with reduced power, which is useful in military applications. On the other hand, reducing transmission power also means a smaller number of feasible links among nodes and, hence, lower connectivity. These two effects have an opposite impact on energy consumption. A large part of recent work on energy efficiency in ad hoc networks is concentrated on energy-efficient routing [164–167], in which the transmitting power level is an additional variable in the routing protocol design [27]. Numerous energy-conscious routing protocols have been proposed. For example, Minimum Power Routing (MPR) selects the path between a given source and destination that will require the least amount of total power expected, while still maintaining an acceptable signal-to-noise ratio at each receiver [89, 91]. It also utilizes physical and link-layer statistics to conserve power, while compensating for the propagation path loss, shadowing, fading, and interference effects.

Energy-efficiency comparison of a number of MAC-layer protocols [26] examines the effectiveness of various media acquisition strategies in the presence of contention. Conventional energy-conserving link-layer protocols are designed for centralized environments where resource-rich base stations are used to control node communication and reduce contention via careful scheduling and traffic buffering [24]. In ad hoc environment, the unpredictable connectivity and limited node buffering capability unfortunately limit the efficacy of these strategies.

Reference [168] points out battery properties that impact on the design of battery powered devices. Power-saving policies at the operating-system level include strategies for CPU scheduling [169, 170] and hard-disk management [171].

It is worth noting that simulation study of the energy consumption of two well-known ad hoc routing protocols [27] running over IEEE 802.11 demonstrated that an energy-oriented performance evaluation may lead to quite different conclusions than a bandwidth-oriented one when judging protocol performance.

At the application level, conventional strategies used to minimize energy consumption for wireless nodes are not applicable to ad hoc networks [29, 31]. Policies that exploit the application semantic or profit by tasks' remote execution have been proposed [156]. By utilizing usage patterns associated with user applications such as e-mail and Web browsing, these techniques reduce energy consumption by letting mobile devices spend as much time as possible in a low-power-consumption sleep state. However, since nodes in ad hoc networks are involved in forwarding other nodes' packets as well, it is difficult to predict the time that a network interface will spend in a low-power sleep state.

1.4.6. Network Security

The wireless and mobile ad hoc nature of MANET brings new security challenges to network design. Because nodes in mobile ad hoc network generally communicate with each other via open and shared broadcast wireless channels, they are more vulnerable to security attacks. In addition, their distributed and infrastructureless nature means that centralized security control is hard to implement and the network has to rely on individual security solutions from each mobile node. Furthermore, as ad hoc networks are often designed for specific environments and may have to operate with full availability even in adverse conditions, security solutions applied in more traditional networks may not be directly suitable [119, 127].

Understanding the possible form of attacks is the first step toward developing good security solutions. In mobile ad hoc networks, the broadcasting wireless medium inherently signifies that an attack may come from any direction and from different layers (network or application transport such as TCP flooding and SYN flooding). Possible attacks include:

- Passive eavesdropping
- Denial-of-service attacks
- Signaling attacks: Attackers may inject erroneous routing information [122] to divert network traffic, or make routing inefficient
- Flow-disruption attacks: Intruders may delay/drop/corrupt all data passing through, but leave all routing traffic unmodified [121]
- Resource depletion attacks: Intruders may send data with the objective of congesting a network or draining batteries [121]
- Data integrity attacks, by accessing, modifying, or injecting traffic
- Stolen device attacks

Several unique solutions have been proposed to address these possible attacks in MANET. Similar to wireline networks, protecting access to wireless network infrastructure is obviously the starting step. Many authentication techniques have been proposed to achieve access control and data integrity. To prevent attackers from injecting erroneous routing information and data traffic, use of digital signatures to authenticate a message has been proposed [120, 122]. Implementation of these schemes requires a certification authority function to manage the private–public keys and to distribute keys via certificates. Since certification authority function is not possible in MANET, this function needs to be distributed over multiple nodes. Reference [120] defines the specific message formats to be used to carry the digital signature. Other methods for access control include the use of the resurrecting duckling technique, in which a mobile device will trust the first device that sends a secret key [128].

Besides authentication, encryption can be used to achieve confidentiality and hide information during transmission or to store information more safely to prevent passive eavesdropping and data integrity attacks via using encryption and decryption keys. But even with encryption, an eavesdropper may be able to identify the traffic pattern in the network and obtain the mode of operation information. Results in [124] suggest that such traffic analysis can be prevented by presenting a constant traffic pattern independent of the underlying operational mode or insertion of dummy traffic.

Other intrusion-related mechanisms proposed include techniques for intrusion-resistant ad hoc routing algorithms (TIARAs) [121] and intrusion detection techniques [123] that enable early detection of intrusion in the network by using intelligent protocols or characteristic "training" data such as rate of change of routing information to identify abnormal media access patterns, or abnormal routing table updates.

1.4.7. Simulation and Performance Evaluation

Simulation plays an important role in MANET technology development. Constructing a real ad hoc network test bed for a given scenario is typically expensive and remains limited in terms of working scenarios, mobility models, etc. Furthermore, measurements are generally non-repeatable. For these reasons, protocol scalability, sensitivity to user mobility patterns, and speeds are difficult to evaluate on a real test bed. Using a simulation or analytic model, on the other hand, permits the study of system behavior by varying all its parameters and considering a large spectrum of network scenarios. For mobile ad hoc network evaluation, simulation modeling is preferred over analytical modeling due to its flexibility and ability to model network-level details. A detailed discussion of methods and techniques for MANETs simulation can be found in [173].

The ability of ad hoc network protocols to correctly behave in a dynamic environment, where device position may continuously change, is a key issue. Therefore, modeling users' movements is an important aspect in ad hoc network simulation. Important aspects that need to be considered [173] during simulation include the definition of the simulated area in which users' movements take place, the rules for modeling users that move beyond the simulated area, the number of nodes in the simulated area, the node mobility model, and the allocation of nodes at the simulation start-up, and so on.

Typically, simulation studies consider a fixed number of users that move inside a closed rectangular area. Rules are defined for users arriving at the edges of the area.

The random waypoint mobility model is the most commonly used technique to define the way users move in the simulated area. According to this model, nodes move according to a broken-line pattern, standing at each vertex for a model-defined pause time (p). Specifically, each node picks a random destination in the rectangular area, samples a speed value according to a uniform distribution in the range (0, $v_{\max}$], and then travels to the destination along a straight line. Once the node arrives at its destination, it pauses for a time p, then chooses (draws) another destination and continues onward. The pause time and the maximum speed, v, are mobility parameters. By changing these values, various system mobility patterns are captured. For example, $p = 0$ signifies that all nodes are always in motion throughout the simulation run.

In order to establish a repeatable simulation environment and make for a valid comparison of results, a set of network environments and performance metrics have been proposed. Common environment metrics used to define the networking context include network size (number of nodes), network density, capacity, connectivity structure (average number of neighbors, transmission range), mobility pattern (speed, range, direction, frequency, etc.), link bandwidth (bps), traffic pattern (packet size, transmission frequency, type of traffic), link characterics (bidirectional or unidirectional), transmission medium (single vs. multichannel), and so on. Commonly used network performance metrics [96, 97] include network settling time, network join time, network depart time, network recovery time, route acquisition time, memory required, maximum number of supported network nodes, frequency of control updates, overhead ratio, number of data packets delivered correctly, energy consumption, percentage of out-of-order delivery, end-to-end data throughput and delay, as well as associated mean, variance, and distribution, and so on.

A very large number of simulation models have been developed to study ad hoc network architectures and protocols under varying network scenarios (number of nodes, mobility rates, etc.) and constraints (bandwidth and energy, latency, throughput, or association stability, etc.). For example, the existence of a large number of routing protocols in MANET areas makes the routing protocol selection a difficult task. Simulation considering the above metrics presents an easier and systematic way to construct the network environment; to collect and analyze the required performance metrics to make fair protocol comparisons in terms of protocol characteristics, functionalities, advantages, drawbacks; and to provide required performance-optimization solutions. Many studies and simulations have been performed to compare and contrast the large number of routing protocols developed for MANETs; see, for example, [56, 175–177]. A theoretical framework is presented in [178] that compares ad hoc network routing protocols (in an implementation-independent manner) by measuring each protocol's performance relative to a theoretical optimum.

Most MANET simulations are carried out using network simulators such as OPNET [179], NS-2 [180], and Glomosim [181] and its commercial version QualNet [182]. These simulators provide advanced simulation environments to test and debug different networking protocols, including collision-detection modules, radio propagation, and MAC protocol. These tools include libraries containing predefined models for most communication protocols (e.g., 802.11, Ethernet, TCP, etc.). In addition, they often provide graphical interfaces that can be used both during the model development phase and during simulation runs to simplify the following dynamic protocol and network behaviors.

Some recent results, however, question the validity of simulations based on these tools. Specifically, [183] presents results of the flooding algorithm using OPNET, NS-2, and Glomosim, as significant disparities between the simulators have been measured recently. The observed differences are not only quantitative (not the same absolute value), but also qualitative (not the same general behavior), making some past observations of MANET simulation studies an open issue.

1.4.8. Quality of Service and Optimization

In mobile ad hoc networks, the presence of additional bandwidth, link and medium constraints, as well as the constant change in network topology, make supporting Quality of Service more difficult than in fixed wireline networks, which only need to deal with static constraints such as bandwidth, memory, or processing power. Due to the lack of sufficiently accurate knowledge of the network states, both instantaneous and predictive, even statistical QoS guarantees may be impossible if the nodes are highly mobile [148]. Consequently, many existing QoS solutions developed for Internet are not suitable for MANET environments and need to be adapted.

In general, Quality of Service is not related to any dedicated network layer, but, rather, requires coordinated efforts from all layers. Important QoS components in the MANET domain include QoS models, QoS Medium Access Control (MAC), QoS routing and resource reservation signaling, and so on [132]. A QoS model outlines the overall Quality of Service goals and architecture for implementing a given application or service. These objectives may include link capacity, latency, link utilization percentage, throughput, bandwidth and energy consumption, and so on. QoS routing refers to the discovery and maintenance of routes that can satisfy QoS objectives under given the resource constraints, whereas QoS signaling is responsible for actual admission control, and scheduling, as well as resource reservation along the route determined by QoS routing or other routing protocols. Both QoS routing and QoS signaling coordinate with the QoS MAC protocol to deliver the QoS service required.

Much research has been done in each of these component areas [130–133]. A first attempt at QoS modeling has been presented in [135], where a Flexible QoS Model for MANET (FQMM) based on both IntServ and Diffserv is proposed. It is a hybrid scheme in that it tries to preserve the per-flow granularity for a small portion of traffic in MANET, while using DiffServ-based per-class granularity for the rest of the traffic.

INSIGNIA is one of the early QoS signaling protocols specifically designed for resource reservation in ad hoc environments [136, 138]. It supports in-band signaling by adding a new option field in the IP header called INSIGNIA to carry the signaling control information. Like RSVP, the service granularity supported by INSIGNIA is per-flow management. The INSIGNIA module is responsible for establishing, restoring, adapting, and tearing down real-time flows. It includes fast-flow reservation, restoration, and adaptation algorithms that are specifically designed to deliver adaptive real-time service in MANETs [136]. If the required resource is unavailable, the flow will be degraded to best-effort service. QoS reports are sent to the source node periodically to report network topology change as well as QoS statistics results such as loss rate, delay, and throughput etc. Other QoS signaling protocols proposed for MANET include dynamic RSVP (dRSVP) [145].

QoS routing helps establish the route for successful resource reservation by QoS signaling in MANETs [132]. This is a difficult task. In order to make an optimal routing decision, QoS routing requires constant updates on link state information such as delay, bandwidth, cost, loss rate, and error rate to make policy decisions, resulting in large amounts of control overhead, which can be prohibitive for a bandwidth-constrained ad hoc environment. In addition, the dynamic nature of MANETs makes maintaining the precise link state information extremely difficult, if not impossible [132, 147, 148]. Finally, even after resource reservation, QoS still cannot be guaranteed due to the frequent disconnection and topology change. Many QoS routing algorithms were published recently. These work with a variety of QoS requirements and resource constraints, for example, CEDAR [140], ticket-based probing [139], predictive location-based QoS routing [149], localized QoS routing [146], and QoS routing based on bandwidth calculation [141].

QoS MAC protocols solve the problems of medium contention, support of reliable unicast communication, and resource reservation for real-time traffic in a distributed wireless environment [132]. Among numerous MAC protocols and improvements that have been proposed, protocols that can provide QoS guarantees to real-time traffic in a distributed wireless environment include the GAMA/PR protocol [142] and the Black-Burst (BB) contention mechanism [144].

1.5. FUTURE RESEARCH DIRECTIONS

Despite the large volume of research activities and recent progress made in the mobile ad hoc networking area both in the research community and industry, there are still many interesting and important research problems to be solved in order to enable the large-scale commercialization of the technology. Future challenges for ad hoc wireless networks include, but are certainly not limited to:

- *Routing Protocol Optimization.* As stated, ad hoc routing has been the most active research area in MANET, and many routing protocols have been proposed, each focusing on solving specific problems in the routing domain and suitable for a specific ad hoc environment. Future research is required to provide a scalable, adaptive, and robust solution that can support commercialization of large ad hoc networks, as well as optimize performance of these protocols for given cost objective, such as energy consumption, throughput, delay, or control overhead. Some promising initiatives in this area include power-aware routing and location-aided routing. Location-aided routing aims at using the mobile node's positioning information provided by GPS or other mechanisms to define associated regions for nodal communication, thereby reducing routing control overhead, saving energy, and improving routing performance. Further work is also needed to investigate the feasibility and performance of hybrid ad hoc routing approaches to allow the integration of hierarchical, table-driven routing protocols with on-demand routing to create scalable routing strategies that can adapt well to various ad hoc environments. Finally, multicasting, which is essential for supporting multiparty multimedia communications, has not been studied as extensively, and significant research effort is needed to come up with efficient multicasting algorithms to cope with multicast group dynamics.
- *QoS Support.* Quality of Service support is inherently difficult in an ad hoc environment. In order to support real-time multimedia applications, effort must be made to control network QoS factors such as end-to-end delay, packet loss, and jitter. It has been recognized that hard QoS guarantees will be difficult to achieve in a dynamic environment. Consequently, there is a trend toward an adaptive QoS approach instead of the "plain" resource reservation method with hard QoS guarantees. Since end-to-end QoS guarantee requires coordinated effort from all layers, more research effort is need to come up with coherent mechanisms that present an "all layer QoS" solution instead of individual optimization within each layer.

- *Simulation.* Modeling and simulation are at present incomplete and inadequately supported tasks. Existing tools do not realistically and cleanly blend the physical and MAC layers with data link, network, and higher layers. More research is needed to address this issue. Most simulation tools in use today do not exceed networks of a hundred nodes in size, and simulation environments that scale to larger networks are only being developed.
- *Security.* The security area has not been addressed adequately in existing research. So far, solutions based on encryption, digital signatures, timestamps, and similar methods, do help in achieving authentication, integrity, nonrepudiation, and privacy to a certain degree, but more work need to be done to identify the possible network trust relationships and physical security requirements as well as to improve proposed algorithms to work in a distributed dynamic environment.
- *Standardization and Interoperability.* Standardization is critical for lowering network and development costs and for ensuring adoption and interoperability. With the ever-increasing number of protocols and algorithms being proposed, guidance is needed in developing solutions that target achieving overall scalability and high system performance, as well as interoperability among the technologies.

In addition to the above areas, further research is needed in the areas of media access control, service discovery, addressing and autoconfiguration, resource management (bandwidth, energy, etc.), location management, billing models, internet protocol operability, and applications for mobile networks, to mention only a few of the most challenging, as MANETs and 4G networking environments in general are taking shape.

1.6. CONCLUSIONS AND INTRODUCTION TO REMAINING CHAPTERS

In coming years, it seems inevitable that mobile computing will flourish and evolve toward integrated, converged fourth generation wireless technology. Ad hoc networking will play an important role in this evolution. Its intrinsic flexibility, ease of maintenance, lack of needed infrastructure, autoconfiguration, self-administration capabilities, and significant cost advantages make it a prime candidate for becoming the stalwart technology for personal pervasive communication. The opportunities for and importance of ad hoc networks are being increasingly recognized by both the research and industry community, as evidenced by the flood of research activities, strong industry interest, and almost exponential growth of the Wireless LAN and Bluetooth sectors. In moving forward and successfully fulfilling this opportunity, developing and seamlessly integrating MANET with other wireless networks and fixed internet infrastructures, the successful addressing of many of the open research and development issues discussed in this article will play a critical role.

The rest of the chapters in this book cover many important areas and design issues in mobile ad hoc networks that, due to space limitations, have been only touched upon in this overview chapter. Specifically, subsequent chapters focus on the following areas:

Chapter 2, entitled "Off-the-Shelf Enablers of Ad Hoc Networks," by Gergely V. Záruba and Sajal K. Das, discusses the WPAN and WLAN technologies as a basis for ad hoc networks. Specifically, the chapter analyzes the IEEE 802.11 family, HiperLAN, and Bluetooth.

Chapter 3, entitled "IEEE 802.11 in Ad Hoc Networks: Protocols, Performance and Open Issues," by Giuseppe Anastasi, Marco Conti, and Enrico Gregori, presents the IEEE 801.11 technology and discusses its utilization for constructing ad hoc networks. Special attention is devoted to the interaction between the TCP protocol and IEEE 802.11-based ad hoc networks. The aim is to analyze the performance of Internet applications such as Web browsing and file transfer in such environments.

Chapter 4, entitled "Bluetooth Scatternet Formation in Bluetooth Networks," by Stefano Basagni, Raffaele Bruno, and Chiara Petrioli, describes the state of the art in scatternet formation using Bluetooth technology, that is, formation of multihop ad hoc nets of Bluetooth devices.

Chapter 5, entitled "Antenna Beamforming and Power Control for Ad Hoc Networks," by Ram Ramanathan, discusses the techniques to guarantee an efficient utilization of channel capacity. These techniques include: (1) utilizing directional antennas in ad hoc networks to increase effective capacity, increase connectivity, and lower probability of detection/interference, and (2) controlling the topology of an ad hoc network by changing the transmitting power.

Chapter 6, entitled "Topology Control in Wireless Ad Hoc Networks," by Xiang-Yang Li, discusses methods for designing and maintaining network topology to enable network scalability, such as how to decide transmission radius to reduce interference and conserve energy while enabling good network connectivity, topology updates, and neighbor discovery.

Chapter 7, entitled "Broadcasting and Activity-Scheduling in Ad Hoc Networks," by Ivan Stojmenovic and Jie Wu, surveys existing methods for broadcasting in a wireless network intelligently (using omnidirectional or directional antennas, with equal or adjusted transmission radii) and for scheduling node activities to ensure both reliability and power and bandwidth efficiency.

Chapter 8, entitled "Location Discovery," by Andreas Savvides and Mani B. Srivastava, surveys the requirements and broad applications that can supported by location discovery, as well as technologies and algorithms that have been developed in this domain, with an emphasis on the application and usage in wireless systems for routing calls to mobile users.

Chapter 9, entitled "Mobile Ad hoc Networks (MANETs): Routing technology for dynamic, wireless networking," by Joseph P. Macker and M. Scott Corson, provides specific insights into standardization activities and efforts involved in mobile ad hoc networking.

Chapter 10, entitled "Routing Approaches in Mobile Ad Hoc Networks," by Elizabeth M. Belding-Royer, presents a comprehensive set of techniques used for routing in ad hoc networks.

Chapter 11, entitled "Energy Efficient Communication in Ad Hoc Wireless Networks," by Laura M. Feeney, summarizes the evaluation of energy consumption in medium-access control, routing, and transport protocols, including metrics and protocols used to prolong network life, and design of localized algorithms that avoid communication overhead for updating network information. Power-efficient medium access and the use of geographic position for power optimization are also discussed.

Chapter 12, entitled "Ad Hoc Network Security," by Pietro Michiardi and Refik Molva, presents recent research in the security area, including recent advances in providing an automated key management scheme that does not require the presence of an external infrastructure or bootstrap phase in which keys are distributed, as well as currently available security mechanisms implemented in the data-link layer.

Chapter 13, entitled "Self-Organized and Cooperative Ad Hoc Networking by Silvia Giordano and Alessandro Urpi, presents methods for exploiting certain characteristics of ad hoc networks (e.g., cooperation and the relationship among nodes) based on community and social network concepts.

Chapter 14, entitled "Simulation and Modeling of Wireless, Mobile, and Ad Hoc Networks," by Azzedine Boukerche and Luciano Bononi, focuses on the use of simulation methods and tools used in the performance analysis of ad hoc network architecture and protocols, including synthetic models for describing the users' mobility and the pros and cons of various available simulation tools (NS-2, Glomosim, etc.).

Chapter 15, entitled "Modeling Cross-Layering Interaction Using Inverse Optimization," by Violet R. Syrotiuk and Amaresh Bikki, discusses modeling protocol interaction at different layers of a networking system.

Chapter 16, entitled "Algorithmic Challanges in Ad Hoc Networks," by Andras Farago, describes solved and open algorithmic problems that form the basis for many of the fundamental solutions and protocols in ad hoc networking.

REFERENCES

1. S. Hara and R. Prasad, "Overview of Multicarrier CDMA," *IEEE Communications Magazine,* Dec. 1997, pp. 126–133.
2. Y. Bing Lin and I. Chlamtac, *Wireless and Mobile Network Architectures,* Wiley, 2000.
3. D-Link WLAN Access Point User Manaul and On-line Help, 2002.
4. H. Harada and R. Prasad, "A New Multi-carrier CDMA/TDD Transmission Scheme Based on Cyclic Extended Spread Code for 4th Generation Mobile Communication System," in *Proceedings of the IEEE International Conference on Personal Wireless Communication Conference,* pp. 319–323, 1997.
5. R. G. T. Anderson, B. Bershad, and D. Wetherall, "A System Architecture for Pervasive Computing," In *Proceedings of the 9th ACM SIGOPS European Workshop,* pp. 177–182, Kolding, Denmark, September 2000.
6. Wireless World Research Forum (WWRF): http://www.ist-wsi.org.
7. I. Chlamtac and M. El-Zarki, "Introduction to Computer Networks," in *Encyclopedia of Telecommunications,* Vol. 9, 1994, Marcel Dekker.
8. S. Giordano, "Mobile Ad-hoc Networks," in *XXX XXX,* Wiley, 2000.
9. C. E. Perkins (Ed.), *Ad Hoc Networking,* Addison-Wesley Longman, 2000.
10. The official Bluetooth web site: http://www.bluetooth.com/.
11. IBM Zurich Research Laboratory web site: http://www.zurich.ibm.com/cs/wireless/bluetooth.html.
12. I. Chlamtac and J. Redi, "Mobile Computing: Challenges and Opportunities," in *Encyclopedia of Computer Science,* 4th Edition, edited by D. Hemmendinger, A. Ralston, and E. Reilly, International Thomson Publishing, 1998.
13. D. L. Lough, T. K. Blankenship, K. J. Krizman, Tutorial on Wireless LANs and IEEE 802. 11, Virginia Polytechnic Institute and State University, http://computer.org/students/looking/summer97/ieee802.htm.

14. L. Goldberg, “Wireless LANs: Mobile Computing’s Second Wave,” *Electronic Design, 26,* June 1995.
15. Y. Bing Lin, Y. R. Huang, A. Pang, and I. Chlamtac, “All-IP Approach for Third Generation Mobile Networks,” *IEEE Network Magazine,* in press.
16. J. A. Freebersyser and B. Leiner, *A DoD Perspective on Mobile Ad Hoc Networks, Ad Hoc Networking,* Addison Wesley, 2001.
17. W. Fifer and F. Bruno, “The Low-Cost Packet Radio,” *Proceedings of the IEEE, 75*(1), 33–42, January 1987.
18. N. Shacham and J. Westcott, “Future Directions in Packet Radio Architectures and Protocols,” *Proceedings of the IEEE 75*(1), 83–99, January 1987.
19. R. Kahn et al. “Advances in Packet Radio Technology,” *Proceedings of the IEEE 66,* 1468–1496, November 1978.
20. B. Leiner, R. Ruth, and A. R. Sastry, “Goals and Challenges of the DARPA GloMo Program,” *IEEE Personal Communications,* 34–43, December 1996.
21. J. Strater and B. Wollman, “OSPF Modeling and Test Results and Recommendations,” Mitre technical report 96W0000017, Xerox Office Products Division, March 1996.
22. C. F. Chiasserini, I. Chlamtac, P. Monti, and A. Nucci, “Optimal Energy Design of Wireless Ad Hoc Networks,” *Lecture Notes in Computer Science (LNCS),* No. 2345, E. Gregori, M. Conti, A. T. Campbell, G. Omidyar, and M. Zukerman, (Eds.), 2002.
23. C.-K. Toh, *Ad Hoc Mobile Wireless Networks: Protocols and Systems,* Prentice Hall PTR, 2002.
24. L. M. Feeney and M. Nilsson, *Investigating the Energy Consumption of a Wireless Network Interface in an Ad Hoc Networking Environment,* Swedish Institute of Computer Science, Kista, Sweden.
25. I. Chlamtac, “Issues in Mobile Computing,” Plenary Address, Seventh IEEE International Conference on Personal, Indoor, and Mobile Radio Communications (PIMRC’96), Taiwan, October 1996.
26. J.-C. Chen, K. M. Sivalingam, P. Agrawal, and S. Kishore, “A Comparison of MAC Protocols for Wireless Local Networks Based on Battery Power Consumption,” in *Proceedings of IEEE Infocom 1998,* San Francisco, CA, March, 1998.
27. L. M. Feeney, “An Energy-consumption Model for Performance Analysis of Routing Protocols for Mobile Ad Hoc Networks,” *Journal of Mobile Networks and Applications,* 2001.
28. J.-H. Chang and L. Tassiulas, “Energy Conserving Routing in Wireless Ad-hoc Networks,” in *Proceedings of IEEE Infocom 2000,* Tel Aviv, Israel, March 2000.
29. P. Gauthier, D. Harada, and M. Stemm, “Reducing Power Consumption for the Next Generation of PDAs: It’s in the Network Interface,” in *Proceedings of MoMuC’96,* September 1996.
30. I. Chlamtac, C. Petrioli, and J. Redi, “Energy-Conserving Access Protocols for Identification Networks,” *IEEE/ACM Transactions on Networking, 7,* 1, February 1999.
31. R. Kravets and P. Krishnan, “Power Management Techniques for Mobile Communication,” in *Proceedings of MobiCom’98,* Dallas, TX, October 1998.
32. P. Krishna, N. H. Vaidya, M. Chatterjee, and D. K. Pradhan, “A Cluster-Based Approach for Routing in Dynamic Networks,” *ACM SIGCOMM Computer Communications Review,* 1997.
33. M. Jiang, J. Li, and Y. C. Tay. “Cluster Based Routing Protocol Functional Specification.” Internet Draft, draft-ietf-manet-cbrp-spec.txt. Work-in-progress.
34. I. Chlamtac, C. Petrioli, and J. Redi, “An Energy-Conserving Access Protocol for Wireless Communications,” in *IEEE International Conference on Communications, ICC’97,* Montreal, Quebec, Canada, June 1997.
35. C.-K. Toh, “Associativity-Based Routing For Ad-Hoc Mobile Networks,” *Journal on Wireless Personal Communications, 4,* First Quarter, 1997.

36. N. H. Vaidya, *Mobile Ad Hoc Networks: Routing, MAC and Transport Issues,* Texas A&M University, 2000.
37. I. Chlamtac and S. Pinter, "Distributed Nodes Organization Algorithm for Channel Access in a Multi-Hop Dynamic Radio Network," *IEEE Transactions on Computers, C-36,* 6, June 1987.
38. S.-L. Wu, Y.-C. Tseng, and J.-P. Sheu, "Intelligent Medium Access for Mobile Ad Hoc Networks with Busy Tones and Power Control," Department of Computer Science and Information Engineering, National Central University, Chung-Li, 32054, Taiwan.
39. P. Karn, "MACA—A New Channel Access Method for Packet Radio," in *ARRL/CRRL Amateur Radio 9th Computer Networking Conference,* pp. 134–140, 1990.
40. V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, "MACAW: A Medium Access Protocol for Wireless LANs," in *Proceedings of SIGCOMM '94,* pp. 212–225, 1994.
41. C. L. Fullmer and J. Garcia-Luna-Aceves, "Floor Acquisition Multiple Access (FAMA) for Packet-Radio Networks," in *Proceedings of SIGCOMM'95,* Nov. 1995.
42. C. R. Lin and M. Gerla, "MACA/PR: An Asynchronous Multimedia Multihop Wireless Network," in *Proceedings of IEEE INFOCOM* '97, Apr. 1997.
43. F. Talucci and M. Gerla, "MACA-BI (MACA By Invitation) A Wireless MAC Protocol for High Speed ad hoc Networking," in Proceedings of ICUPC'97, Nov. 1997.
44. I. Chlamtac and A. Farago, "Making Transmission Schedules Immune to Topology Changes in Multi-Hop Packet Radio Networks," *IEEE/ACM Transactions on Networking, 2,* 1, February 1994.
45. Z. J. Hass and J. Deng, "Dual Busy Tone Multiple Access (DBTMA): Performance Evaluation," in *49th Annual International Vehicular Technology Conference,* Oct. 1998.
46. J. Deng and Z. J. Hass, "Dual Busy Tone Multiple Access (DBTMA): A New Medium Access Control for Packet Radio Networks," in *Proceedings of the IEEE International Conference on Universal Personal Communications,* pp. 314–319, 1998.
47. IEEE Std. 802.11-1997: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications. Institute of Electrical and Electronics Engineers, Inc., New York, 1997.
48. C. L. Fullmer and J. J. Garcia-Luna-Aceves, "Floor Acquisition Multiple Access for Packet Radio Networks," in *SIGCOMM'95,* pp. 262–273, ACM, 1995.
49. J. J. Garcia and C. L. Fullmer, "Performance of Floor Acquisition Multiple Access in Ad-hoc Networks," in *Proceedings of the Third IEEE Symposium on Computers and Communications,* pp. 63–68, 1998.
50. S. Khanna, S. Sarkar, I. Shin, An Energy Measurement Based Collision Resolution Protocol, University of Pennsylvania.
51. F. A. Tobagi and L. Kleinrock, "Packet Switching in Radio Channels: Part II—The Hidden Terminal Problem in Carrier Sense Multiple Access Modes and the Busy Tone Solution," *IEEE Transactions in Communications, 23*(12), 1417–1433, 1975.
52. S. Singh and C. S. Raghavendra, "Power Efficient MAC Protocol for Multihop Radio Networks," in *International Symposium on Personal, Indoor and Mobile Communications,* 1998.
53. E. K. Wesel, *Wireless Multimedia Communicaions: Networking Video, Voice, and Data,* Addison-Wesley, 1998.
54. S.-L. Wu, Y.-C. Tseng, and J.-P. Sheu, *Intelligent Medium Access for Mobile Ad Hoc Networks with Busy Tones and Power Control,* Technical Report NCU-HSCCL-1999-02, Department of Computer Science and Information Engineering, National Central University, May 1999.
55. E. D. Kaplan (Ed.), *Understanding GPS: Principles and Applications,* Artech House, 1996.
56. J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva, "A Performance Comparison of Multihop Wireless Ad Hoc Network Routing Protocols," in *Proceedings of the Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'98),* pp. 85–97, 1998.

57. E. M. Royer and C.-K. Toh, “A Review of Current Routing Protocols for Ad-Hoc Mobile Wireless Networks,” *IEEE Personal Communications,* April 1999

58. V. D. Park and M. S. Corson, “A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks,” in *Proceedings of INFOCOM ‘97,* April 1997.

59. C. E. Perkins and P. Bhagwat, “Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers,” *Computer Communications Review,* 234–244, October 1994.

60. C. E. Perkins, E. M. Royer, and S. R. Das, “Ad hoc on-demand distance vector (AODV) routing,” IETF Internet draft, draft-ietf-manet-aodv-11. txt, June 2002.

61. C. E. Perkins and E. M. Royer, “Ad-hoc On-Demand Distance Vector Routing,” in *Proceedings of 2nd IEEE Workshop on Mobile Computing Systems and Applications,* February 1999.

62. L. R. Ford Jr. and D. R. Fulkerson, *Flows in Networks,* Princeton University Press, 1962.

63. C. -C. Chiang, H. K. Wu, W. Liu, and M. Gerla, “Routing in Clustered Multihop, Mobile Wireless Networks with Fading Channel,” in *Proceedings of IEEE SICON’97,* pp. 197–211, April 1997.

64. S. Murthy and J. J. Garcia-Luna-Aceves, “An Efficient Routing Protocol for Wireless Networks,” *ACM Mobile Networks and Applications Journal,* Special Issue on Routing in Mobile Communication Networks, pp. 183–197, October 1996.

65. D. B. Johnson and D. A. Maltz, *Dynamic Source Routing in Ad-Hoc Wireless Networks, Mobile Computing,* edited by T. Imielinski and H. Korth, Kluwer Academic Publishers, pp. 153–181, 1996.

66. C.-K. Toh, “A Novel Distributed Routing Protocol to Support Ad-Hoc Mobile Computing,” in *Proceedings of the 1996 IEEE Fifteenth Annual International Phoenix Conference on Computers and Communication,*pp. 480–486, March 1996.

67. H. Hassanein, *Load-Aware Routing in Wireless Ad Hoc Networks,* Department of Computing and Information Science, Queens University.

68. R. Droms, “Dynamic host configuration protocol, RFC 2131,” http://www.ietf.org/rfc/rfc2131.txt, Mar 1997.

69. M. Günes and J. Reibel, “An ip address configuration algorithm for zeroconf. mobile multihop ad-hoc networks,” in *Proceedings of the International Workshop on Broadband Wireless Ad-Hoc Networks and Services,* Sophia Antipolis, France, September 2002. ETSI.

70. C. Petrioli, S. Basagni, and I. Chlamtac “Configuring BlueStars: Multihop Scatternet Formation for Bluetooth Networks,” *IEEE Transactions on Computers,* (to be published in the June 2003 issue).

71. S. Thomson and T. Narten, “Ipv6 stateless address autoconfiguration,” rfc 2462. http://www.ietf.org/rfc/rfc2462.txt, December 1998.

72. K. Fall and K. Varadhan. *The ns Manual,* Nov 2000.

73. M. Günes and O. Spaniol, Routing Algorithms for Mobile Multi-Hop Ad-Hoc Networks,

74. P. Jacquet, P. Muhlethaler, and A. Qayyum, “Optimized Link State Routing Protocol,” Internet Draft, draft-ietf-manet-olsr-00.txt, November 1998.

75. G. Pei, M. Gerla, and T.-W. Chen. “Fisheye State Routing in Mobile Ad Hoc Networks,” in *Proceedings of the 2000 ICDCS Workshops,* pp. D71–D78, Taipei, Taiwan, April 2000.

76. L. Kleinrock and K. Stevens, “Fisheye: A Lenslike Computer Display Transformation,” Technical Report, UCLA, Computer Science Department, 1971.

77. G. Pei, M. Gerla, and X. Hong, “LANMAR: Landmark Routing for Large Scale Wireless Ad Hoc Networks with Group Mobility,” in *Proceedings of IEEE/ACM MobiHOC 2000,* pp. 11–18, Boston, MA, August 2000.

78. P. F. Tsuchiya, “The Landmark Hierarchy: A New Hierarchy for Routing in Very Large Networks,” *Computer Communications Review, 18,* 4, 35–42, August 1988.

79. T.-W. Chen and M. Gerla, “Global State Routing: A New Routing Scheme for Ad-hoc Wireless Networks,” in *Proceedings of IEEE ICC’98,* http://www.ics.uci.edu/~atm/adhoc/paper-collection/gerla-gsr-icc98.pdf.

80. Z. J. Haas and M. R. Pearlman, “The Zone Routing Protocol (ZRP) for Ad Hoc Networks,” November 1997.

81. M. Joa-Ng and I.-T. Lu, “A Peer-to-Peer Zone-Based Two-Level Link State Routing for Mobile Ad Hoc Networks,” *IEEE Journal on Selected Areas in Communications,* Special Issue on Ad-Hoc Networks, Aug. 1999, 1415–1425.

82. A. Iwata, C.-C. Chiang, G. Pei, M. Gerla, and T.-W. Chen, “Scalable Routing Strategies for Ad Hoc Wireless Networks,” *IEEE Journal on Selected Areas in Communications,* Special Issue on Ad-Hoc Networks, Aug. 1999, 1369–1379.

83. Mingliang J. and Jinyang Li, Y. C. Tay, “Cluster Based Routing Protocol,” August 1999, IETF Draft, http://www.ietf.org/internet-drafts/draft-ietf-manet-cbrp-spec-01.txt

84. R. Dube et al., “Signal Stability Based Adaptive routing for Ad Hoc Mobile Networks,” *IEEE Personal Communications,* Feb. 1997, pp. 36–45.

85. V. D. Park and M. S. Corson, “A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks,” in *Proceedings of INFOCOM’97,* Apr. 1997.

86. P. Kuosmanen, “Classification of Ad Hoc Routing Protocols,” Finnish Defence Forces, Naval Academy, Finland, petteri.kuosmanen@mil.fi.

87. S. Basagni, I. Chlamtac, V. R. Syrotiuk, and B. A. Woodward, “A Distance Routing Effect Algorithm for Mobility (DREAM),” in *Proceedings of the Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking, MobiC’98,* Dallas, TX, October, 1998.

88. E. L. Madruga, J. J. Garcia-Luna-Aceves, “Scalable Multicasting: The Core-Assisted Mesh Protocol,” 1999.

89. M. W. Subbarao, “Dynamic Power-Conscious Routing for MANETs: An Initial Approach,” in *Proceedings of IEEE VTC Fall 1999,* Amsterdam, The Netherlands, 1999.

90. I. Chlamtac and O. Weinstein, “The Wave Expansion Approach to Broadcasting in Multi-Hop Radio Networks,” in *Proceedings of IEEE INFOCOM,* San Francisco, California, April 1987.

91. M. W. Subbarao, “Mobile Ad Hoc Data Networks for Emergency Preparedness Telecommunications—Dynamic Power-Conscious Routing Concepts,” 2000.

92. Chlamtac and S. Kutten, “On Broadcasting in Radio Networks—Problem Analysis and Protocol Design,” *IEEE Transactions on Communications, COM-33,* 12, December 1985.

93. R. Sivakumar, P. Sinha, and V. Bharghavan, “CEDAR: a Core-Extraction Distributed Ad hoc Routing Algorithm,” *IEEE Journal on Selected Areas in Communications, 17,* 8, August 1999.

94. S. Basagni, I. Chlamtac, and A. V. R. Syrotiuk, “Location Aware One-to-Many Communication in Mobile Multi-hop Wireless Networks,” in *Proceedings of IEEE Vehicular Technology Conference (VTC),* Tokyo, Japan, May 2000.

95. L. M. Feeney, “A Taxonomy for Routing Protocols in Mobile Ad Hoc Networks,” SICS Technical Report T99/07, October 1999, http://www.sics.se/~lmfeeney/research.html.

96. M. W. Subbarao, Ad Hoc Networking Critical Features and Performance Metrics,” 1999.

97. M. S. Corson and J. Macker, “Mobile Ad Hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations,” *RFC 2501, IETF,* January 1999.

98. R. de Oliveira and O. Braun, “TCP in Wireless Mobile Ad Hoc Networks,” Technical Report, IAM-02-003, July 2002

99. N. H. Vaidya, “TCP for Wireless and Mobile Hosts,” *MobiCom’99 Tutorial,* Texas A&M University, 1999

100. H. Balakrishnan, V. Padmanabhan, and R. Katz, “The Effects of Asymmetry on TCP Perfor-

mance," in *Proceedings of Third ACM/IEEE Mobicom Conference,* Budapest, Hungary, September 1997.

101. D. Sun and H. Man, "Performance Comparison of Transport Control Protocols over Mobile Ad Hoc Networks."
102. G. Holland and N. H. Vaidya, "Analysis of TCP performance over Mobile Ad Hoc Networks," in *International Conference on Mobile Computing and Networking (MOBICOM),* August 1999.
103. F. Wang and Y. Zhang, "Improving TCP Performance over Mobile Ad-Hoc Networks with Out-of-Order Detection and Response."
104. M. Gerla, K. Tang, and R. Bagrodia, "TCP Performance in Wireless Multi-Hop Networks," in *Proceedings of IEEE Workshop on Mobile Computing Systems and Applications (WMCSA),* pp. 41–50, February 1999.
105. M. Gerla, R. Bagrodia, L. Zhang, K. Tang, and L. Wang, *TCP over Wireless Multi-Hop Protocols: Simulation and Experiments,* Computer Science Department, University of California at Los Angeles
106. Z. Fu, P. Zerfos, K. Xu, H. Luo, S. Lu, L. Zhang, and M. Gerla, "On TCP Performance in Multihop Wireless Networks," in *Proceedings of InfoCom,* 2003.
107. N. Vaidya, M. Mehta, C. Perkins, and G. Montenegro, "Delayed Duplicate Acknowledgements: A TCP-Unaware Approach to Improve Performance of TCP Over Wireless," Tech. Rep. 99-003, Computer Science Department, Texas A&M University, February 1999.
108. A. Bakre and B. Badrinath, "I-TCP: Indirect TCP for Mobile Hosts," in *Proceedings of 15th International Conference on Distributed Computing Systems (ICDCS),* pp. 136–143, May 1995.
109. A. V. Bakre and B. R. Badrinath, "Implementation and Performance Evaluation of indirect TCP," *IEEE Transactions on Computers, 46,* March 1997.
110. H. Balakrishnan, S. Seshan, E. Amir, and R. Katz, "Improving TCP/IP Performance over Wireless Networks," in *Proceedings of the 1st ACM International Conference on Mobile Computing and Networking (MOBICOM),* November 1995.
111. K. Brown and S. Singh, "M-TCP: TCP for Mobile cellular Networks," in *Proceedings of the ACM SIGCOMM Computer Communication Review,* pp. 19–43,1997.
112. V. Tsaoussidis and H. Badr, "TCP-Probing: Towards an Error Control Schema with Energy and Throughput Performance Gains," in *Proceedings of the 8th IEEE International Conference on Network Protocols,* 2000.
113. T. Goff, J. Moronski, and D. Phatak, "Freeze-TCP: A True End-to-End Enhancement Mechanism for Mobile Environments." in *Proceedings of INFOCOM,* 2000.
114. C. Zhang and V. Tsaoussidis, "TCP Real: Improving Real-time Capabilities of TCP over Heterogeneous Networks," in *Proceedings of the 11th IEEE/ACM NOSSDAV 2001,* New York, 2001.
115. B. S. Bakshi, P. Krishna, D. K. Pradhan, and N. H. Vaidya, "Improving Performance of TCP over Wireless Networks," in *Proceedings of International Conference on Distributed Computing Systems,* May 1997.
116. H. Balakrishnan, V. Padmanabhan, S. Seshan, and R. Katz, "A Comparison of Mechanisms for Improving TCP Performance over Wireless Links," in *ACM SIGCOMM,* Stanford, CA, August 1996.
117. Z. Haas and P. Agrawal, "Mobile-TCP: An Asymmetric Transport Protocol Design for Mobile Systems," in *Proceedings of ICC'97,* Montreal, Canada, June 1997.
118. R. Yavatkar and N. Bhagwat, "Improving End-to-End Performance of TCP over Mobile Inter-Networks," in *Proceedings of Workshop on Mobile Computing Systems and Applications,* December 1994.

119. V. Kärpijoki, *Security in Ad Hoc Networks,* Helsinki University of Technology, Telecommunications Software and Multimedia Laboratory.
120. S. Jacobs and M. S. Corson, “Manet authentication architecture,” IETF MANET Working Group Internet Draft, August 1998.
121. R. Ramanujan, A. Ahamad, J. Bonney, R. Hagelstrom, and K. Thurber, “Techniques for Intrusion-Resistant Ad Hoc Routing Algorithms (TIARA),” in *Proceedings of MILCOM,* October 2000.
122. L. Zhou and Z. J. Haas, “Securing Ad Hoc Networks,” *IEEE Network Magazine, 13,* 6, November/December 1999
123. Y. Zhang and W. Lee, “Introduction detection in wireless ad-hoc networks,” in *Proceedings of Annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM),* August 2000.
124. S. Jiang, N. H. Vaidya, and W. Zhao, “Routing in Packet Radio Networks to Prevent Traffic Analysis,” in *Proceedings of IEEE Information Assurance and Security Workshop,* West Point, NY, June 2000.
125. V. Kärpijoki, “Signalling and Routing Security in Mobile Ad Hoc Networks,” in *Proceedings of the Helsinki University of Technology Seminar on Internetworking—Ad Hoc Networks,* Spring 2000.
126. J. Moy, “Security Architecture for the Internet Protocol.” RFC 2401, November 1998, Internet Society.
127. S. Mäki, “Security Fundamentals in Ad Hoc Networking,” in *Proceedings of the Helsinki University of Technology Seminar on Internetworking—Ad Hoc Networks,* Spring 2000.
128. F. Stajano, and R. Anderson, “The Resurrecting Duckling: Security Issues for Ad-hoc Wireless Networks,” in *Proceedings of the 7th International Workshop on Security Protocols,* Lecture Notes in Computer Science, Springer-Verlag, 1999.
129. J. Kong et al., “Providing Robust and Ubiquitous Security Support for MANET,” 2001.
130. I. Chlamtac and A. Lerner, “Fair Algorithms for Maximal Link Activiation in Multi-Hop Radio Networks,” *IEEE Transactions on Communications, COM-35,* 7, July 1987.
131. S. R. Das, R. Castaneda, and J. Yan, “Comparative Performance Evaluation of Routing Protocols for Mobile Ad hoc Networks,” http://www.ececs.uc.edu/~sdas/pub.html.
132. K. Wum and J. Harms, “QoS Support in Mobile Ad Hoc Networks,” *Crossing Boundaries, 1,* 1, Fall 2001.
133. X. Luo, B. Li, I. Thng, Yi-Bing Lin, and I. Chlamtac, “An Adaptive Measured-Based Pre-Assignment Scheme with Connection-level QoS Support for Mobile Networks,” *IEEE Transactions on Wireless Communications,* in press.
134. M. S. Corson and A. T. Campbell, “Towards Supporting Quality of Service in Mobile Ad hoc Networks.”
135. H. Xiao, W. K. G. Seah, A. Lo, and K. C. Chua, “A Flexible Quality of Service Model for Mobile Ad-Hoc Networks,” in *Proceedings of IEEE VTC2000-spring,* Tokyo, Japan, May 2000.
136. S.-B. Lee and A. T. Campbell, “INSIGNIA: In-band Signaling Support for QoS in Mobile Ad Hoc Networks,” in *Proceedings of 5th International Workshop on Mobile Multimedia Communications (MoMuC, 98),* Berlin, Germany, October 1998.
137. I. Chlamtac and A. Lerner, “Link Allocation in Mobile Radio Networks with Noisy Channel,” in *Proceedings of IEEE INFOCOM,* Bar Harbour, Florida, April 1986.
138. G.-S. Ahn, A. T. Campbell, S.-B. Lee, and X. Zhang, “INSIGNIA,” Internet Draft, draft-ietf-manet-insignia-01.txt, Oct. 1999.
139. S. Chen and K. Nahrstedt, “An Overview of Quality-of-Service Routing for the Next Generation High-Speed Networks: Problems and Solutions,” *IEEE Networks,* Special Issue on Transmission and Distribution of Digital Video, Nov./Dec. 1998.

140. P. Sinha, R. Sivakumar, and V. Bharghavan, “CEDAR: a Core-Extraction Distributed Ad Hoc Routing algorithm,” in *Proceedings of IEEE Infocom’99,* New York, NY, March 1999

141. R. Lin and J. S. Liu, “QoS Routing in Ad Hoc Wireless Networks,” *IEEE Journal on Selected Areas in Communications,* August 1999.

142. Muir and J. J. Garcia-Luna-Aceves, “An Efficient Packet-Sensing MAC Protocol for Wireless Networks,” *ACM Journal on Mobile Networks and Applications, 3,* 2, 221–234, August 1998.

143. J. Liu, B. Li, T. Hou, and I. Chlamtac, “On Optimal Layering and Bandwidth Allocation for Multi-Session Video Broadcasting,” *IEEE Transactions on Wireless Communication,* in press.

144. J. L. Sobrinho and A. S. Krishnakumar, “Quality-of-Service in Ad Hoc Carrier Sense Multiple Access Wireless Networks,” *IEEE Journal on Special Areas in Communications, 17,* 8, August 1999.

145. M. Mirhakkak, N. Schult, and D. Thomson, *Dynamic Quality-of-Service for Mobile Ad Hoc Networks,* The MITRE Corporation, 2000.

146. X. Yuan and A. Saifee, *Path Selection Methods for Localized Quality of Service Routing,* Department of Computer Science, Florida State University.

147. S. Chen and K. Nahrstedt, *An Overview of Quality-of-Service Routing for the Next Generation High-Speed Networks: Problems and Solutions,* Department of Computer Science, University of Illinois at Urbana-Champaign.

148. B. Li, *QoS-aware Adaptive Services in Mobile Ad-hoc Networks,* Department of Electrical and Computer Engineering, University of Toronto.

149. S. H. Shah and K. Nahrstedt, “Predictive Location-Based QoS Routing in Mobile Ad Hoc Networks,” fshshah.

150. H. Zhu, G. Zeng, and I. Chlamtac, “Control Scheme Analysis for Multimedia Inter- and Intra-Stream Synchronization,” in *Proceedings of IEEE International Conference on Communications (ICC 2003),* Anchorage, Alaska, May 2003.

151. The IEEE 802.16 Working Group Web Site: http://www.ieee802.org/16.

152. F. A. Tobagi and L. Kleinrock, “Packet Switching in Radio Channels: Part II—The Hidden Terminal Problem in Carrier Sense Multiple Access Modes and the Busy Tone Solution,” *IEEE Transactions on Communications, 23*(12), 1417–1433, 1975.

153. Special issue, “Energy-Aware Ad Hoc Wireless Networks,” *IEEE Wireless Communications, 9,* 4, August 2002.

154. G. H. Forman and J. Zahorjan, “The Challenges of Mobile Computing,” *IEEE Computer, 27,* 4, 38–47, 1994.

155. J. R. Lorch and A. J. Smith, “Software Strategies for Portable Computer Energy Management,” *IEEE Personal Comunications, 5,* 3, 60–73, 1998.

156. C. Jones, K. Sivalingam, P. Agarwal, and J. C. Chen, “A Survey of Energy Efficient Network Protocols for Wireless and Mobile Networks,” *ACM/Kluwer Wireless Networks (WINET), 7,* 4, 343–358, 2001.

157. M. Rulnick and N. Bambos, “Mobile Power Management for Wireless Communication Networks,” *ACM/Baltzer Wireless Networks, 3,* 1, 1996.

158. M. Zorzi and R. R. Rao, “Energy Constrained Error Control for Wireless Channels,” in *Proceeding of IEEE GLOBECOM ‘96,* London, UK, November, pp. 1411–1416, 1996.

159. R. Bruno, M. Conti, and E. Gregori, “Optimization of Efficiency and Energy Consumption in p-Persistent CSMA-Based Wireless LANs,” *IEEE Transactions on Mobile Computing, 1.* 1, 10–31, 2002.

160. C. Raghavendra and S. Singh, “PAMAS: Power Aware Multi-Access Protocol with Signaling for ad hoc networks,” *ACM Computer Communication Review* (July 1998) 5–26.

161. L. Feeney, “Energy Efficient Communication in Ad Hoc Wireless Networks,” in *Mobile Ad*

Hoc Networking, S. Basagni, M. Conti, S. Giordano, and I. Stojmenovic (Eds.), IEEE Press–Wiley, 2004.

162. H. Woesner, J. P. Ebert, M. Schlaeger, and A. Wolisz, "Power-Saving Mechanisms in Emerging Standards for Wireless LAN's: The MAC-Level Perspective," *IEEE Personal Communications* (Special Edition on Power Saving), *5,* 3, 40–48, 1998.
163. J. P. Ebert, B. Stremmel, E. Wiederhold, and A. Wolisz, "An Energy-efficient Power Control Approach for WLANs," *Journal of Communications and Networks (JCN), 2,* 3, 197–206, 2000.
164. V. Rodoplu and T. H.-Y. Meng, "Minimum Energy Mobile Wireless Networks," *IEEE Journal on Selected Areas in Communications, 17*(8), 1333–1344, August 1999.
165. S. Singh, M. Woo, and C. S. Raghavendra, "Powerware Routing in Mobile Ad Hoc Networks," in *Proceedings of ACM/IEEE Mobicom,* Oct. 1998, pp. 181–90.
166. I. Stojmenovic and X. Lin, "Power-Aware Localized Routing in Wireless Networks," in *Proceedings of IEEE Symposium on Parallel and Distributed Processing Systems,* May 2000.
167. R. Ramanathan and R. Rosales-Hain, "Topology Control of Multi-Hop Wireless Networks Using Transmit Power Adjustment," in *Proceedings of IEEE INFOCOM,* Tel Aviv, Israel (March 2000).
168. C.-F. Chiasserini and R. R. Rao, "Pulsed Battery Discharge in Communication Devices," in *Proceedings of MOBICOM 1999,* pp. 88–95.
169. J. R. Lorch and A. J. Smith, " Scheduling Techniques for Reducing Processor Energy Use in MacOS," *ACM/Baltzer Wireless Networks, 3,* 5, 311–324, 1997.
170. M. Weiser, B. Welch, A. Demers, and S. Shenker, "Scheduling for Reducing CPU Energy. USENIX Association," in *First Symposium on Operating System Design and Implementation,* Monterey, CA, pp. 13–23, 1994.
171. D. P. Helmbold, D. E. Long, and B. Sherrod, "A Dynamic Disk Spin-Down Technique for Mobile Computing," in *Proceedings of the Second Annual ACM International Conference on Mobile Computing and Networking,* NY, pp. 130–142, 1996.
172. M. Weiser, "The Computer for the Twenty-First Century," *Scientific American,* September 1991.
173. L. Bononi and A. Boukerche, "Simulation and Modeling of Wireless, Mobile and Ad Hoc Networks," in *Mobile Ad Hoc Networking,* S. Basagni, M. Conti, S. Giordano, and I. Stojmenovic (Eds.), IEEE–John Wiley, 2004.
174. G. Holland and Nitin H. Vaidya, "Analysis of TCP Performance over Mobile Ad Hoc Networks," *Wireless Networks 8*(2–3), 275–288, 2002.
175. P. Johansson, T. Larsson, N. Hedman, and B. Mielczarek, "Routing Protocols for Mobile Ad-Hoc Networks—A Comparative Performance Analysis," in *Proceedings of ACM Mobicom '99,* August 1999, pp. 195–206.
176. S. R. Das, R. Castaneda, and J. Yan, "Simulation Based Performance Evaluation of Mobile Ad Hoc Network Routing Protocols," *ACM/Baltzer Mobile Networks and Applications (MONET) Journal,* 179–189, July 2000.
177. Samir R. Das, Charles E. Perkins, and E. M. Royer, "Performance Comparison of Two On-demand Routing Protocols for Ad Hoc Networks," in *Proceedings of INFOCOM 2000,* Tel Aviv, Israel, March 2000.
178. A. Farago and V. Syrotiuk, "MERIT: A Scalable Approach for Protocol Assessment," *ACM/Kluwer MONET, 8,* 5, Oct. 2003, Special issue on "Mobile Ad Hoc Network," A. Campbell, M. Conti, and S. Giordano (Eds.).
179. OPNET Modeler, http://www.opnet.com/products/modeler/home.html.
180. The Network Simulator—ns-2, http://www.isi.edu/nsnam/ns/index.html.
181. *GloMoSim,* Global Mobile Information Systems Simulation Library, http://pcl.cs.ucla.edu/projects/glomosim/.

182. Qualnet simulator, http://www.qualnet.com/.

183. D. Cavin, Y. Sasson, and Andrè Schiper, "On the Accuracy of MANET Simulators," in *Proceedings of ACM POMC'02,* Toulouse, France, October 2002,

184. Special Issue on "Mobile Ad Hoc Networking," *Cluster Computing Journal, 5,* 2, April 2002.

185. A. Savvides and M. Srivastava, "Location Discovery" in *Mobile Ad Hoc Networking,* S. Basagni, M. Conti, S. Giordano, and I. Stojmenovic (Eds.), IEEE–Wiley, 2004.

186. J. Widmer, M. Mauve, H. Hartenstein, and H. Füßler, "Position-Based Routing, in Ad-Hoc Wireless Networks," M. Ilyas (Ed.), *The Handbook of Ad Hoc Wireless Networks,* CRC Press, 2002.

187. S. Giordano, I. Stojmenovic, and L. Blazevic, "Position Based Routing Algorithms for Ad Hoc Networks: A Taxonomy," in *Ad Hoc Wireless Networking,* X. Cheng, X. Huang, and D. Z. Du (Eds.), Kluwer, 2003.

188. S. Giordano and I. Stojmenovic, "Position based ad hoc routes in ad hoc networks," Chapter 16 in *The Handbook of Ad Hoc Wireless Networks,* M. Ilyas (Ed.), CRC Press, 2003.

189. Y.-C. Tseng and C.-H. Hsu, "Location-Aware Routing and Applications of Mobile Ad Hoc Networks," Chapter 18 in *The Handbook of Ad Hoc Wireless Networks,* M. Ilyas (Ed.), CRC Press, 2003.

190. H. Takagi and L. Kleinrock, "Optimal Transmission Ranges for Randomly Distributed Packet Radio Terminals," *IEEE Transactions on Communications, 32,* 3, 246–257, 1984.

191. T. C. Hou and V. O. K. Li, "Transmission range control in multihop packet radio networks," *IEEE Transactions on Communications, 34,* 1, 38–44, 1986.

192. E. Kranakis, H. Singh, and J. Urrutia, "Compass Routing on Geometric networks," in *Proceedings of 11th Canadian Conference on Computational Geometry,* Vancouver, August, 1999.

193. Y. B. Ko and N. H. Vaidya, "Location-Aided Routing (LAR) in Mobile Ad Hoc Networks," in *Proceedings of MOBICOM* 1998, pp. 66–75; *Wireless Networks, 6,* 4, 307–321, July 2000.

194. W. H. Liao, Y. C. Tseng, and J. P. Sheu, "GRID: A Fully Location-Aware Routing Protocol for Mobile Ad Hoc Networks," in *Proceedings of IEEE HICSS,* January 2000; *Telecommunication Systems, 18,* 64–84, 2001.

195. L. Blazevic, L. Buttyan, S. Capkun, S. Giordano, J.-P. Hubaux, and J.-Y. Le Boudec, "Self-organization in Mobile Ad Hoc Networks: The Approach of Terminodes," *IEEE Communication Magazine,* 166–175, June 2001.

196. I. Chlamtac, C. Petrioli, and J. Redi, "Energy-Conserving Selective Repeat ARQ Protocols for Wireless Data Networks," in *Proceedings of IEEE 9th International Symposium on Personal, Indoors and Mobile Radio Communications, PIMRC,* Boston, Mass, September, 1998.

197. S. Sajama and Z. Haas, "Independent Tree Ad Hoc Multicast Routing (ITAMAR)," *Special Issue on Mobile Ad Hoc Networks,* ARM/Kluwer (MONET), *8,* 5, 2003.

198. L. Ji and S. M. Corson, "Explicit Multicasting for Mobile Ad Hoc Networks," *Special Issue on Mobile Ad Hoc Networks,* ARM/Kluwer (MONET), *8,* 5, 2003.

CHAPTER 2

OFF-THE-SHELF ENABLERS OF AD HOC NETWORKS

GERGELY V. ZÁRUBA and SAJAL K. DAS

2.1 INTRODUCTION

Today, when ad hoc networking professionals or would-be professional talk about ad hoc networks, they almost always implicitly assume that these networks are based on one of the wireless local area network (WLAN) technologies. The majority of research papers published on simulation-based performance evaluation of proposed ad hoc routing protocols assume underlying WLAN medium access control (MAC) and physical (PHY) layers. Most recently, with the appearance of short-range wireless personal area networking (WPAN) technologies, researchers also started to use the characteristics of these technologies as a basis for underlying transport assumptions to evaluate their novel network (or higher-) layer protocols.

It is extremely important to point out, that WLANs and WPANs are significantly different from ad hoc networks. Ad hoc networks have received their name due to the fact that there is no predefined structure or infrastructure of communication over which they should be established, but they consist of nodes that relay information to their neighbors possibly on behalf of other neighbors. Ad hoc networks are often called wireless multihop networks due to the fact that most packets will have to be relayed by several nodes before they reach their destinations. WLANs, on the other hand, are based on infrastructure—just like cellular networks—where there are dedicated access points (likely connected to the wired infrastructure) controlling their entire transmission range, namely their wireless domain. WLANs are considered single-hop networks, since all nodes attached to the access point talk to only the access point, which is the only entity equipped with a routing function. Fortunately, as outlined in the next subsection, the histories and requirements for ad hoc and

Mobile Ad Hoc Networking. Edited by Basagni, Conti, Giordano, and Stojmenovic.
ISBN 0-471-37313-3 © 2004 Institute of Electrical and Electronics Engineers, Inc.

WLAN/WPAN technologies are converging, and most (if not all) technologies defined for WLANs/WPANs are extended to be employable as the basis for ad hoc networking.

2.1.1 The Converging History of Ad Hoc Networks and WLANs

The idea of both WLANs and ad hoc networks date back to approximately the same time, the early 1970s. Although the main driving force behind ad hoc networks was the need for survivable, infrastructureless and hard-to-detect military applications, WLANs received a lot of attention from academia and companies interested in commercial deployment.

In 1972, the Department of Defense (DoD) initiated a new program on Packet Radio Networks (PRNET) with the intention to create technologies for the battlefield that do not need a previously deployed infrastructure but are highly survivable even when some of the radios fail or are destroyed. The medium-access technology employed was a slightly modified version of the ALOHA protocol developed two years earlier in academia to interconnect the computing infrastructure over four Hawaiian islands with eight transceivers. Thus, the first ad hoc network was already using wireless LAN technology as the underlying MAC and PHY layers. Later on, in the early 1980s, the PRNET program was replaced by the Survivable Adaptive Radio Networks (SURAN) program, improving upon the physical properties and routing of PRNET. Technologies to create moderate-cost ad hoc networks outside of the DoD were not present, and since there were very few mobile devices with any computing power, there was no need for commercial deployment either.

In the early 1990s, mobile computing power became affordable for the masses in the forms of laptops, notebooks, and personal digital assistants (PDAs). At the same time, hardware and software, especially open-source software, became widely available for trivial interconnection of computers and maybe connection to the emerging global network, the Internet. It was just a question of time of when the need for mobile connectivity would reach a critical mass to be worthy for commercial companies to look into developing standards, technologies, and products to enable mobile, i.e., wireless interconnection of devices. The early 1990s was also the time of the renaissance of ad hoc networking research, wherein packet radio networks were renamed ad hoc networks [23, 36], and old ad hoc networking problems became important research topics again. There was a commercial need for mobile interconnection, leading toward a push for wireless infrastructure based standards as well as a strong lobbying from research organizations to develop technologies that could be used as the basis of ad hoc networking (with more stress on the former). Due to the major interest from several companies, the Institute of Electrical and Electronics Engineers (IEEE) 802 Group in charge of computer communication networks established a subcommittee, IEEE802.11, to standardize and unify techniques and technologies to be used for wireless LANs. Since the subcommittee was established involving experts from companies and academia, it was also aware of the need for infrastructureless communications and was working in parallel to address both infrastructure-based and infrastructureless needs.

The DoD never lost interest in ad hoc networking, and funded programs such as the Global Mobile Information Systems (GloMo) and Near-term Digital Radio (NTDR), the former addressing Ethernet-type connectivity, and the latter focusing on military applications (NTDR also became the first nonprototype, real ad hoc network in the world). By 1997, the IEEE802.11 subcommittee had approved its first WLAN standard, defining the physical layer as well as the MAC and logical-link control layers for infrastructured and infrastructureless communication.

Today, the prices for IEEE802.11-based technologies are within everybody's reach and since an infrastructureless mode is defined, it has become the premier choice for the underlying bottom two layers (PHY and MAC) for most simulation, test-bedding, and even commercial ad hoc networks and applications. Yet, one should not forget that it is the infrastructureless part of the specification that permits the ad hoc mode, not the WLAN technology, which provides for ad hoc networks. Another factor to keep in mind is that most of the revenues are generated from the technology being deployed in WLANs and, thus, some protocol issues significantly different in WLAN and ad hoc scenarios will show a strong bias toward a primary WLAN behavior.

2.1.2 Wireless LANs

In the strict sense of the word, WLANs are infrastructure-based wireless networks, in which there is a need to deploy wireless access points ahead of time; these access points control network usage in their respective transmission range or domain. A local area network's spatial span is usually between 10 meters to a few hundred meters; thus, the same coverage range is demanded from a wireless LAN. A node that wants to connect wirelessly to a WLAN, should (i) be in the transmission range of the access point, (ii) obtain or carry an IP address from the same IP domain (assuming IP communication) that the access point is in, and (iii) use the access point as a bridge or router for every packet it sends or receives.

Wireless bandwidth is one of the most important natural resources of countries; thus, its usage is regulated by national regulation bodies. In the United States, the regulatory body in charge of the national radio frequency resources is the Federal Communications Commission (FCC). In order for a frequency band to be used, the FCC has to issue licenses to devices using that band as well as a license to operate devices in that band. The FCC has designated several frequency bands, commonly known as the ISM (Industrial, Scientific, and Medical) and/or U-NII (Unlicensed-National Information Infrastructure) bands, for which an FCC license is only needed for the device and not for the usage of the band. WLANs take advantage of these ISM bands, so the operators do not have to request permits from the regulatory bodies. The most common ISM bands for WLANs in order of their importance are: 2.4 GHz–2.483 GHz, 5.15 GHz–5.35 GHz, 5.725 GHz–5.825 GHz (United States) and 5.47 GHz–5.725 GHz (European Union), and 902 MHz–928 MHz (not relevant).

Since WLANs rely on a centrally controlled structure, just like cells of cellular networks, several access points can be used to create cellular-like WLAN structures. Some WLAN technologies are more suited for such large-coverage, cellular-like WLANs, whereas others may not perform well in such scenarios as it will be pointed out later in this chapter. The term *hot-spot* recently became a frequently used term, referring to an area covered by one or more WLAN access points to provide Internet connectivity at a fraction of the cost of a cellular data connection to users whose terminals are equipped with wireless network interface cards. Providing hotspots is an extremely controversial issue; current cellular providers are likely to loose revenue unless they are the ones providing the service.

2.1.3 Wireless PANs

The term wireless personal area networks came along with the appearance of its first representative technology: Bluetooth. WPANs (or, in short, PANs) are very short range wireless networks with a coverage radius of a few centimeters to about 10 meters, connecting devices in the reach of individuals, thus receiving the name. WPANs do not necessarily require an infrastructure; they imply single-hop networks in which two or more devices are connected in a point-to-multipont "star" fashion. Although the communication distance is shorter, so that the power requirements are lessened, Bluetooth provides a significantly lower symbol rate than WLANs. Fortunately, this contradicting "feature" is currently being addressed and it is likely that future WPAN technologies will provide users with options of significantly higher transmission speeds.

2.1.4 Digital Radio Properties

In order to fully comprehend the different aspects of medium-access control in WLAN and WPAN standards and specifications, it is necessary to possess basic knowledge on the behavior/terminology of digital radio transmissions. If using radio as the medium for communication, the bit error rate (BER) due to undesired interfering sources could become 8–10 orders of magnitude higher than in an optical or wired medium. The attenuation of radio signals is proportional to at least the square of the distance and the square of the carrier frequency in open propagation environments, in which there are no obstacles (not even the earth's surface) reflecting the radio signal, and the receiver and transceiver are in line of sight. In real environments, statistically, the received signal strength can be decaying with as much as the fourth power of the transmitter–receiver distance, due to obstacles absorbing and reflecting the radio signal. Additionally, in a mobile environment, reflection and absorption of signals from obstacles causes fading effects that can be classified into short- and large-scale fadings depending on how far the transmitter moves away from the transmitter.

Rayleigh fading describes the fading of the signal when the transmitter–receiver distance varies around the wavelength of the carrier signal (about 12 cm at 2.4 GHz). With Rayleigh fading, one has to consider that a radio signal can be received through different paths via obstacles reflecting the signal. Signals received from multiple paths travel different distances; thus, their phases can vary significantly at the receiver, causing amplification and attenuation of each other. Rayleigh fading causes local signal strength minima, or *fading dips,* that are about half a wavelength (about 6.25 cm at 2.4 GHz) away from each other, strongly depending on the carrier frequency.

Log-normal fading describes the fading effect when the signal strength's variation is measured on a large-scale (much greater than the wavelength of the carrier) movement. With log-normal fading, different reflecting and line-of-sight components' strengths can vary with the order of the sizes of obstacles (buildings, etc.) absorbing the energy of the signal; log-normal fading dips are thus 2–3 orders of magnitude farther away than those of Rayleigh fading.

Thus, the received signal strength does not only depend on the approximate distance from the transmitter, but strongly depends on the exact distance (see Figure 2.1) and location, and on the exact frequency carrier used; that is, it is possible to produce a significant change in the received signal strength just by moving the receiver a few centimeters or by changing the carrier frequency by a few kilohertz.

Time dispersion is yet another problem to address—signals bouncing back from obstacles have a time shift comparable to the duration of bit times. Time dispersion could cause the reception of contradicting information, called *intersymbol interference* (ISI).

Since transmission and reception cannot occur at the same time on the same frequency at a single node, and because most building blocks of receivers and transmitters

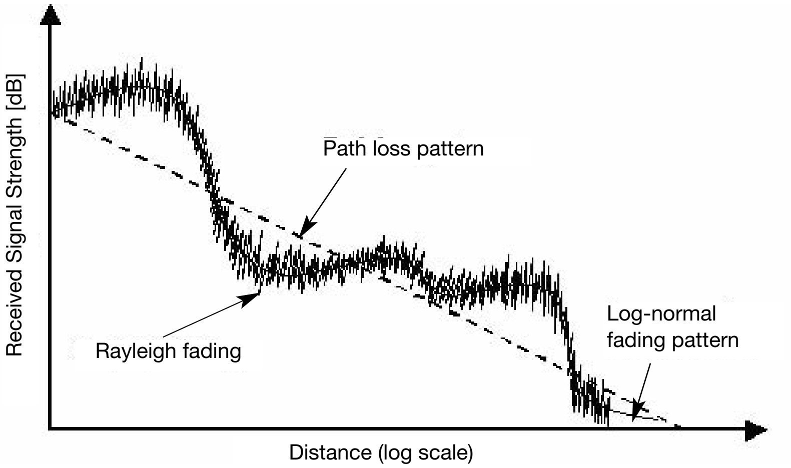


Figure 2.1. Received signal strength.

are the same, it makes economic sense to use time-division duplexing to provide only a single radio unit per device that can be switched between reception and transmission modes. Additionally, the consecutive reception and transmission events of the radio do not necessarily have to take place at the same frequency carrier in order to reduce the risk of being in a fading dip. Unfortunately, it takes significant time (up to a few hundred microseconds) to switch radios between transmission and reception modes (with or without changing the frequency), waiting for all the transients to settle. This time is sometimes referred to as *radio switch-over time* or *radio turn-over time,* during which the radio is useless.

Since radio frequency is a scarce resource, it needs to be used wisely. In order to increase the capacity of a system, the same frequency (band) may be reused at some distance where the other signal becomes low. In order to reduce the reuse distance, thus reducing the interference, systems are sometimes required to implement radio-power control, with which the transmission power to different clients can be dynamically adjusted depending on the reading of the received signal strength.

To reduce the average transmission energy over small frequency bands and to provide better protection against fading dips, *spread-spectrum* (SS) technologies are employed (in fact, the FCC requires SS to be used in the ISM bands). The most well known and widely used SS technologies are (Fast) Frequency Hopping (FH or FFH), Direct Sequence Spreading (DSS), and the novel Orthogonal Frequency Division Multiplexing (OFDM) and Ultra Wide Band (UWB). With FFH, the frequency band is divided up into several narrower bands (using a central carrier frequency in each of these narrow bands). An FFH transmission will use one of the narrow bands for a short period of time, then switch to another, and, again, another, cyclically. The time spent at each carrier frequency is called the *dwell time*. In DSS, the signal to be transmitted is multiplied by a high-speed chip-code or pseudorandom noise (PN) sequence, essentially spreading the energy of the signal over a larger band (resulting in less spectral efficiency). With OFDM, just like with FFH, several frequency carriers are defined, but, unlike FFH, more than one carrier may be used at the same time to transmit different segments of the data. As will be shown, Bluetooth employs FFH, whereas IEEE802.11b and IEEE802.11a employ DSS and OFDM, respectively, and UWB is in its infancy.

The reader interested in more details of digital radio signal propagation and fading effects and their mitigation is referred to [39, 40]. Additionally, [44] provides a good overview of differences in propagation for TDMA/FDMA and CDMA systems.

The rest of this chapter is organized as follows: Section 2.2 introduces WLAN technologies and outlines why they can/cannot be used for ad hoc networking. Section 2.3 deals with WPAN technologies, focusing mostly on Bluetooth, and outlines the problems researchers are facing before Bluetooth can be used for ad hoc networking. Section 2.4 concludes the chapter.

2.2 WIRELESS LAN TECHNOLOGIES

As described in the previous section, the history of WLANs starts with the ALOHA system. In the early 1990s, radio technologies became mature enough to enable the production of relatively inexpensive digital wireless communication interfaces. The first generation of WLANs operated in the 900 MHz ISM band, with symbol rates of around 500 kbps, but they were exclusively proprietary, nonstandard systems, developed to provide wireless connectivity for specific niche markets (e.g., military or inventorying). The second-generation systems came along around 1997, enjoying a strong standardization effort. They operated in the 2.4 GHz range and provided symbol rates of around 2 Mbps. The IEEE802.11 Working Group (WG) and its similarly named standard were the most successful of the standardization efforts. People did not have to wait long for an inexpensive third-generation (2.4 GHz band, 11 Mbps symbol rate) WLAN standard and equipment, as the IEEE802.11b Task Group (TG) was quick in standardizing it, and, due to increased need, products rolled out extremely quickly. Although the IEEE802.11a TG was formed at the same time as IEEE802.11b TG and its standard was available at approximately the same time, it took longer for the first IEEE802.11a products to appear. IEEE802.11a operates in the 5.2 GHz band with speeds up top 54 Mbps (or 108 Mbps in a non-standardized “turbo” or dual mode) and represents the fourth generation of WLANs. The Wireless Ethernet Compatibility Alliance (WECA) was established by companies interested in manufacturing IEE802.11b and IEEE802.11a products. WECA forged the by now widely accepted term Wi-Fi (Wireless Fidelity) to replace the user-unfriendly IEEE802.11 name. WECA is known today as the Wi-Fi Alliance and provides certification for 2.4 GHZ and 5.2 GHz products based on the IEEE802.11b and IEE802.11a standards, respectively.

While the IEEE802.11 WG was working on IEEE’s WLAN standard, the European Telecommunication Standards Institute (ETSI) was working on another standard known as HiperLAN (High Performance Radio LAN). HiperLAN was released at about the same time as the first IEEE802.11 standard in 1998 but has received less attention due to its more stringent manufacturing requirements (representing better qualities, too). HiperLAN operates in the 5.2 GHz band with data rates up to 20 Mbps. The ETSI updated the HiperLAN standard in 2000, releasing HiperLAN 2, which provides similar data rates as IEEE802.11a while enabling easy architectural integration into 3G wireless networks (UMTS) and providing quality of service (QoS) provisioning.

In this section the readers will be introduced to the standardization efforts of the different IEEE802.11 Task Groups as well as to the technology of HiperLAN 1 and 2. It will be shown how these standards provide for not only WLAN usage scenarios but also for ad hoc networking.

2.2.1 IEEE802.11 Technological Overview

The IEEE802.11 Working Group was formed in 1990 to define standard physical (PHY) and medium-access control (MAC) layers for WLANs in the publicly available ISM bands. The original goal was to have data rates of 2 Mbps, falling back to 1 Mbps in the presence of interference or if the signal became too weak. Originally, three different physical layer options were provided: (i) infrared, (ii) frequency hopping spread spectrum (FHSS) at 2.4 GHz, and (iii) direct sequence spread spectrum (DSSS) at 2.4 GHz. Due to the possible need, two kinds of operation modes were also defined: a client-server, regular WLAN mode that received the name IM-BSS (Infrastructure Mode Basic Service Set), and an ad hoc operational mode called IBSS (Independent Basic Service Set). A Basic Service Set (BSS) is nothing but a group of at least two nodes or *stations* (STA) cooperating via the wireless interface.

The infrared PHY layer did not catch up and has been neglected subsequently. The FHSS PHY used 79 different carrier frequencies with 22 different hopping patterns, defining 22 virtual channels with a dwell time of 20 ms (50 hops/s). Although most of the research comparing the DSSS PHY and the FHSS PHY showed that the interference resistance and resilience of the FHSS PHY layer was superior, the FHSS PHY slowly lost the interest of the IEEE80.11 group and more emphasis was put on the DSSS PHY, mainly due to the fact that increasing the rate was hardly possible using the FHSS PHY. The DSSS PHY divided up the available 80 MHz band at the 2.4 GHz range into three nonoverlapping channels, each of them having around 20 MHz of bandwidth, thus enabling interinterferenceless operation of three different networks in the same spatial area. The 1 or 2 Mbps stream was used to modulate a so-called Baker sequence—a well-defined PN (pseudo random noise) sequence to spread the information over the respective 20 MHz band. The original MAC and PHY specifications of the IEEE802.11 were released in 1997.

Two different MAC channel access methods were defined. The first method, Distributed Coordination Function (DCF) to be used ether in the Infrastructure Mode or in the IBSS ad hoc mode employing the Carrier Sense Multiple Access–Collision Avoidance (CSMA/CA) MAC protocol, was first proposed in [25]. The second (optional) access method is the Point Coordination Function (PCF), to be solely used in the Infrastructure Mode, based on a MAC polling scheme. Only a few products have the capability to work with a PCF method, and since the PCF is not defined for the ad hoc mode further description of it is omitted in this chapter.

According to the IEEE802.11 standard, all stations (STA) have to be able to work with the DCF. The goals of the 802.11 group were to provide a similar service on the radio interface as the interface defined for wired LANs in the IEEE802.3 standard or Ethernet; that is, best effort with high probability access but no QoS guarantees. The IEEE802.11 MAC protocol is described in Chapter 3 of this book together with analyses of its performance in ad hoc environments.

Providing security was a major concern of the IEEE802.11 group, whose goal was to provide at least the same level of security as the wired Ethernet. IEEE802.11 defines its own privacy protocol called WEP—Wired Equivalent Privacy. Since in IEEE802.11 packets are broadcast over radio, it is relatively easy to intercept messages and to get attached to a network. Detecting access points is relatively easy even when they do not broadcast their so-called SSID periodically (which they do more often than not), and since most of the access points provide access to a network with a DHCP server, attaching to foreign networks is a relatively easy process for hackers. WEP was supposed to provide an optional encryption service in the MAC layer to enable the communication between access points and clients that share the same secret key. With WEP enabled, the MAC layers will encode each IEEE802.11 frame before transmission with an RC4 cipher (by RSA Security) using a 40, 64, or 128 (WEP-2) bit key and a pseudorandom 24 bit number, whereas the other side will decode the same stream using the same key and random number. The random number is used to increase the lifetime of the key, yet it has been shown that in a busy network, just by listening to the channel for a while, keys can be easily decoded if the original shared key remains the same [6, 9].

In the rest of this section, readers will be introduced to the IEEE802.11 variants (Task Groups), starting with the most popular IEEE802.11b or Wi-Fi 2.4 GHz, and continuing with the strongly emerging IEEE802.11a or Wi-Fi 5.2 GHz. Some insight will be provided into the soon-to-be approved IEE802.11g and the other Task Groups' work (e.g., TGs c, d, e, f, g, and h).

2.2.1.1 IEEE802.11b (Wi-Fi 2.4 GHz). The goal of Task Group b was to increase the maximum bit rate in the 2.4 GHz frequency range while maintaining interoperability with the original standard. The standard was released in 1999, keeping the original MAC layer but redefining the PHY layer to only work with DSSS, thus increasing the spectral efficiency of the three channels with bit rates of up to 11 Mbps each (with fall-back rates of 5.5, 2, and 1 Mbps). It did not take long for Wi-Fi to become widely accepted throughout the world for corporate WLANs, wireless home networks, and so-called hotspots at airports and cafés, as well as by the ad hoc networking community as an easy-to-set up basis for ad hoc testbeds.

2.2.1.2 IEEE802.11a (Wi-Fi 5.2 GHz). Although Task Groups a and b were established at the same time and the standards were accepted at the same time, IEEE802.11a products did not arrived on the market until late 2001 due to technological difficulties. The goal of Task Group a was to port IEEE802.11 to the newly available U-NII at 5.2 GHz and to provide higher bit rates. Thus, the original MAC layer was kept and the PHY was reworked to provide rates of up to 54 Mbps (with fall-back rates of 48, 36, 24, 18, 12, 9, and 6 Mbps). Since the available band at U-NII is about 300 MHz, eight nonoverlapping bands were defined; thus, eight different IEEE802.11a-based WLAN networks can operate in the same space without interference. This is essential to build cellular kinds of structures, in which neighboring cells should not use the same frequency (to reduce interference). With eight different bands (compared to three with IEE802.11b), it becomes relatively easy to establish noninterfering cellular structures. DSSS was not efficient at working with these high bit rates while satisfying frequency regulatory specifications, so a new spectrum spreading technology called OFDM (Orthogonal Frequency Division Multiplexing) or COFDM (Code OFDM) was accepted. OFDM was specifically developed for indoor environments, addressing indoor-specific fading effects.

With OFDM, the signal to be transmitted is modulated over several frequency carriers. In IEEE802.11a, a 20 MHz bandwidth channel is divided into 52 subcarriers, each about 300 kHz wide; 48 of these subcarriers are used as carriers for the data, whereas the remaining four are employed for forward-error correction. Modulation is performed by changing the phase and amplitude of each of the subcarriers. To provide different symbol rates, different levels of amplitudes and phase shift keying are employed (e.g., binary phase shift keying, 16-level shift keying, etc.).

Although the power attenuation due to distance is at least four times as much at the 5.2 GHz range than at the 2.4 GHz range, and signal energy is more likely to be absorbed by obstacles, it has been shown by researchers at Atheros Communications [13]—a pioneer of IEE802.11a products—that the performance of IEEE802.11a is superior to the performance of IEEE802.11b at distances less than 70 meters, by at least a factor of two (see Figure 2.2). Due to this fact and due to the availability of eight channels, IEEE802.11a is likely going to have a prosperous future. Equipment manufactured by some companies extends the standard by introducing even higher-rate modes capable of transmitting with a 108 Mbps symbol rate.

2.2.1.3 IEEE802.11g. Task Group g is working on an extension to IEEE802.11b at 2.4 GHz, enabling transmission at symbol rates of 54 Mbps while retaining the fall-back speeds of IEEE802.11b, thus ensuring interoperability. After a long and rough debate, Task Group g has agreed to the adoption of OFDM technology (while keeping DSS for the interoperability mode); the standard is expected to be finalized at the end of 2002. Although IEEE802.11g-based equipment will provide the same symbol rate as IEEE802.11a, it will still have the same three-channel restriction of the original standard as well as it will operate in the crowded 2.4 GHz range.

All of the previously outlined IEEE802.11-based technologies can be used and deployed as the PHY and MAC layers of ad hoc networks.

2.2.1.4 Other IEEE802.11 Task Groups

IEEE802.11h. There were strong European concerns that 802.11a could interfere with NATO satellites and microwave radar systems. To avoid such interference, two extensions

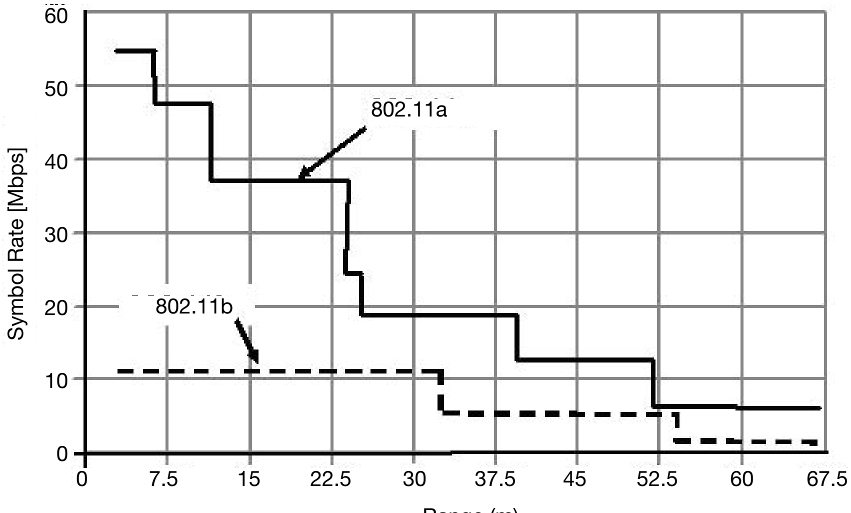


Figure 2.2. Symbol rates of IEEE802.11b versus IEEE802.11a [13].

to the PHY of 802.11a were added in 802.11h, one of them being the capability to select the employed channel automatically based upon observations (DFS—Dynamic Frequency Selection), the other ensuring the enforcement of strict radio power control (TPC—Transmit Power Control).

IEEE802.11e. Task Group e is addressing the flaw of IEEE802.11, working in a best-effort mode but not being able to provide with any QoS provisioning. This Task Group is redefining both the centrally controlled channel access as well as redefining the contention-based channel access of CSMA/CA, including priorities to ensure that packets with higher priorities enjoy access benefits comparable to lower-priority packets in a Differentiated Services manner. This later function is called the *Enhanced Distributed Coordination Function* (EDCF).

IEEE802.11c is a wireless extension to IEEE802.1D, enabling bridging using IEEE802.11 (irrelevant to ad hoc networking).

IEEE802.11d deals with including country-specific information into the beacon transmissions, so STAs are informed of what part of the spectrum is available and what radio constraints they have to obey to (e.g., maximum transmission power).

IEEE802.11f is defining a standard interaccess-point communication protocol for users roaming between access points (irrelevant to ad hoc networks).

IEEE802.11i addresses the flaws of WEP, improving the wireless security at the MAC layer.

2.2.1.5 Further Reading. The reader interested in more high-level details is referred to the 802.11-Planet [3], an online resource on IEEE802.11-related information and news. Readers looking for a more detailed description can obtain the freely available IEEE 802 standards [1, 2] as a result of a new initiative of IEEE 802 to increase interoperability of devices. For a brief online explanation of the OFDM principles, the reader is referred to McCormick's tutorial [26] or to the online white papers and materials of the OFDM Forum [34].

2.2.2 HiperLAN 1 and 2

HiperLAN [16] is the well-known name of the WLAN standardization efforts of the European Telecommunications Standards Institute (ETSI); more precisely, it is being developed by the BRAN (Broadband Radio Access Networks) project of ETSI. HiperLAN 2 [17] is the new version of the standard, providing more bandwidth and interoperability considerations with third-generation wireless networks (e.g., Universal Mobile Telecommunication System or UMTS).

HiperLAN 1 is defined to work in the 5.2 GHz U-NII band, providing symbol rates of up to 23.5 Mbps. Unfortunately, HiperLAN was not picked up by any companies to manufacture products—it quickly became obsolete. ETSI-BRAN has proposed HiperLAN 2, hoping for better acceptance.

The PHY layer of HiperLAN 2 is nearly identical to that of IEEE802.11h (which is a European-initiated extension to IEEE802.11a), using OFDM as the basis. The main difference between HiperLAN2 and IEEE802.11a lies in the definition of the MAC layer. As IEEE802.11a relies on a CSMA/CA-based channel access related to Ethernet, HiperLAN 2 is based on a TDMA approach, with scheduling principles taken from Wireless ATM. HiperLAN 2 thus is able to provide QoS provisioning and can be used for guaranteed real-time data delivery. The MAC layer of HiperLAN 2 defines both a centralized (infrastructure) mode and an ad hoc mode, similarly to IEEE802.11.

Since no company has yet manufactured inexpensive, commercially available HiperLAN products, there are no ad hoc network testbeds based on HiperLAN. The 802.11 standards seem to be more widely accepted than HiperLAN 1 or 2, despite the advertised superiority of HiperLAN 2. Just as with HiperLAN 1, there are no products currently available in large quantities for HiperLAN 2 hindering its deployment as the basis for ad hoc networks. Ad hoc routing protocols (and their simulation) relying on HiperLAN have been proposed [14, 19] but not as widely as protocols relying on IEEE802.11 standards. Optimized Link State Routing (OLSR) [14] is specifically tailored toward HiperLAN. The reader interested in more details is referred to the standards [16, 17] or the excellent white papers provided at the HiperLAN2 Global forum [24].

2.2.3 Infrared WLANs

Although not mentioned yet, the commercial history of WLANs began in 1979 with the Diffused Infrared WLAN project of IBM in Switzerland. The main disadvantage of using photonic electromagnetic waves is that light requires line-of-sight transmission—the receiver and transmitter have to be physically visible to each other. Although fixed environments can be engineered to abide by the line-of-sight rules, mobility can render an infrared WLAN useless. Omnidirectionality of transmissions is not achievable since light is absorbed by most conventional obstacles (such as furniture, the computing unit itself, or people). Due to these major disadvantages, infrared transmission has never taken off as a WLAN competitor (e.g., the original 802.11 defines the operation on an infrared medium as well). It is rarely even used for short-range wireless connections, despite the fact that many portables are equipped with an IrDA (Infrared Data Association) port.

Using infrared transmission in ad hoc networks would defeat the purpose of the ad hoc requirements—networks have to work in all kinds of (mostly hostile) environments. Yet there are projects (such as [12, 22]) exploiting the inexpensive infrared technology for a limited population of ad hoc nodes in indoor environments where the diffusion of the signal can be used as a benefit to somewhat overcome the problem of obstacles.

2.2.4 UWB

Ultra Wide Band (UWB) [39] is a novel spread-spectrum technique acknowledged by the FCC in Spring 2002. UWB can be used for communication as well as to “see through walls,” thus its commercial usage is strongly restricted by the FCC, making it a short-to-medium range wireless communication technology. UWB does not use conventional frequency carriers but generates very short duration rectangular pulses (close to that of Dirac pulses), thus spreading the energy of the transmission over an extremely wide spectrum. Due to this extreme spreading of the energy, UWB does not pose a significant interfering source at any band, and it does not require line of sight.

The first UWB chips have just appeared on the market but it will take a tremendous amount of additional research and standardization effort until UWB-based network adapters become commercially available. UWB has all the properties needed to be the next most popular PHY layer for ad hoc networks. The 802.15.3 Group is also considering UWB as the basis for a high-speed WPAN standard.

2.2.5 Using IEEE802.11 for Ad Hoc Networking

As mentioned earlier, Wi-Fi is extremely popular among ad hoc network researchers as an off-the-shelf support for their simulation or testbedding efforts. In this subsection, some Wi-Fi-based simulation libraries and testbeds will be outlined.

Most major network-simulation toolkits have either an integrated or a contributed IEEE802.11 library. The three most widely used simulators for ad hoc networks—NS2 [33], OPNET [35], and GloMoSim [18]—come with their own implementation of the MAC and PHY layers of IEEE802.11. By far the most simulation efforts of ad hoc routing protocols are carried out assuming (and employing) IEEE802.11-based MAC and PHY layers of one of the above simulation tools.

Due to the availability of inexpensive Wi-Fi products that can be used to establish ad hoc networks, it would be more of a challenge to list all projects that have established an ad hoc network testbed than to list those universities and research labs that do not have any. Here, some of the major projects are listed, starting with possibly the most well-known public license testbed. Uppsala University in Sweden provides everybody the opportunity to build their own Wi-Fi-based ad hoc testbed by providing a GNU Public License on their Ad Hoc Protocol Evaluation (APE) Testbed [5]. APE aims to make the establishment of ad hoc testbeds as easy as possible while providing all the functions required for customization. Project MART (Mobile Ad Hoc Routing Testbed) [30] at the Helsinki University of Technology is establishing a college-wide Wi-Fi-based ad hoc network to evaluate different proposed ad hoc routing protocols.

The MONARCH Project [32] uses a Wi-Fi-enabled ad hoc testbed to evaluate the Dynamic Source Routing (DSR) ad hoc routing approach proposed by them. They also provide the functionality to connect the ad hoc network to a traditional IP network using gateways. The MOMENT Lab at the University of California, Santa Barbara, has its own Wi-Fi-based testbed [31], running on pocket PCs, laptops, and desktops, to evaluate their proposed ad hoc routing protocol: AODV (Ad Hoc On Demand Distance Routing). A project in the R&D Group of Acticom [4] is focused on an ad hoc routing testbed to research multimedia-aware routing protocols for ad hoc networks. The testbed is based on the Wi-Fi 2.4 GHz technology (to be extended to Wi-Fi 5.2 GHz), using multimedia-enabled laptops and running video conferencing applications over their ad hoc network. The Wireless Network Testbed (WNT) [42] at the University of Surrey, United Kingdom, focuses on the evaluation of mobility management protocols, QoS provisioning techniques, routing, and reconfigurability with their Wi-Fi-based ad hoc network. Trinity College in Dublin, Ireland, envisions a Wi-Fi-based ad hoc network covering the entire city of Dublin, using their DAWN (Dublin Ad Hoc Wireless Network) testbed [15]. DAWN is not only envisioned as a testbed but also as the ad hoc medium for fourth-generation (4G) wireless systems, and is fully operational on the campus. Unfortunately, as pointed out in the next paragraph, Wi-Fi was not designed to serve multihop networks, and the community has yet to produce an inexpensive ad hoc tailored PHY and MAC standard.

An extensive analysis of the problems related to the use of IEE802.11 in ad hoc networks is presented in Chapter 3. Here, we would like to point out in advance that Wi-Fi has not been developed for ad hoc networking and, thus, it can exhibit undesired behavior when used for ad hoc networking. Although IEEE802.11 was developed keeping an ad hoc mode in mind, this ad hoc mode is tailored toward simple point-to-point connections; that is, to interconnect laptops for quick file transfers without the buffering and relaying requirement of access points. A recent article [43] in the *IEEE Communication Magazine* points out the shortcomings of the IEEE802.11 MAC layer in providing for ad hoc networks. In [43] the authors claim that the Wi-Fi MAC does not suit ad hoc networks well and that Wi-Fi-based ad hoc testbeds will not perform properly and may cause significant secondary problems (such as TCP instability and unfairness between nodes), reducing the effectiveness of the proposed routing protocols.

2.3 WIRELESS PAN TECHNOLOGIES

Wireless Personal Area Networks (WPANs) are short to very short range (less than 10 meters) wireless networks covering the immediate surroundings of individuals. WPAN technologies are not (and should not be) considered to be contenders of WLAN technologies, but are destined to complement WLANs. The market segment of WPANs is different from that of WLANs; not only is the required range shorter but the required service levels are also different. A PAN is the next wireless networking paradigm in the ordered list of WAN-MAN-LAN paradigms. To enable the embedding of WPAN technologies into general, low-cost devices, theses technologies have to have small footprints, very low costs, and relaxed power requirements. WPAN technology can be used, for example, to interconnect portable computers/digital assistants and their peripherals, to connect sensors/actuators, to connect devices worn by individuals establishing *personal operating spaces* (POS), or to connect devices in cars without the need for cabling. Cost effectiveness is the major keyword that one should associate with WPANs.

2.3.1 Short History

The term personal area network was forged and its standardization started by the establishment of an "Ad Hoc Group" within the IEEE Portable Applications Standards Committee (PASC). In 1998, a Study Group inside the 802.11 Working Group was formed to develop a project authorization request. In March 1999, the 802.15 Working Group was established. Meanwhile, industrial interest groups were formed throughout the world to address the same low-range, low-power, low-cost networking needs. The HomeRF working group/consortium was formed in March 1998, focusing on the home environment—a larger domain than personal area but smaller than local area, with needs similar to PANs. The Bluetooth Special Interest Group (SIG) was formed in May 1998 with the goal of defining an industry standard to replace short-range data cables. Bluetooth took the same route as the IEEE WPAN working group (strong overlap in interested parties), overtaking the IEEE efforts, whereas HomeRF was getting more and more away from WPAN.

The first publicly released version of the Bluetooth specification of the Bluetooth SIG became available in the fourth quarter of 1999 but, due to disturbing imperfections, a new version was released in February 2001. Meanwhile, the IEEE802.15 working group had formed four Task Groups and a Study Group for different WPAN requirements. Task Group 1 (805.15.1) adopted the bottom layers of the Bluetooth specification in June 2002, whereas Task Groups 2, 3, and 4 and the Study Group are concentrating on coexistence with WLANs, and high-rate, low-rate, and alternative-high-rate versions of the standard.

2.3.2 Bluetooth Technological Overview

The Bluetooth SIG was formed in May 1998 by the so-called promoter companies, consisting of Ericsson, IBM, Intel, Nokia, and Toshiba, and later on 3Com, Lucent, Microsoft, and Motorola. The SIG also contains associate members; participating entities pay membership fees and, in turn, can vote or propose modifications for the specifications to come. Adopter companies can join the SIG for free but can only access the oncoming specifications if these have reached a given evolutional level.

The name Bluetooth supposedly comes from a Scandinavian history-enthusiast engineer involved in the early stages of developing and researching this short-range technology, and the name stuck; nobody being able to propose a better one. Bluetooth was the nickname for Harold Blåtand—"Bluetooth,"—King of Denmark (940–985 A.D.). Bluetooth conquered both Norway and Denmark, uniting the Danes and converting them to Christianity. One of the major goals of the Bluetooth standard is to unite the "communication worlds" of devices, computers, and peripherals and to convert "the wired" into wireless; thus, the analogy.

The protocol stack of Bluetooth is depicted in Figure 2.3. Bluetooth is designed so that a single chip can implement the bottom three layers with a serial (RS-232, USB, or similar) interface connecting the chip to the controller host through the so-called HCI (Host Controller Interface).

2.3.2.1 The RF Layer. The physical or RF Layer (Radio Frequency) of Bluetooth is built on a synchronous fast-frequency-hopping paradigm with a symbol rate of 1 Mbps operating in the publicly available 2.4 GHz ISM band. In a normal operation mode, Bluetooth units will change the carrier frequency (hop) 1600 times a second over 79 different carrier frequencies separated 1 MHz apart, starting with 2.402 GHz. (Since the 2.4 GHz ISM band is not equally available in all countries, e.g., France and Spain, Bluetooth enables the operation on a reduced band with only 23 different carrier frequencies.) The modulation scheme employed is similar to that of GSM, that is, GFSK (Gaussian Fre-

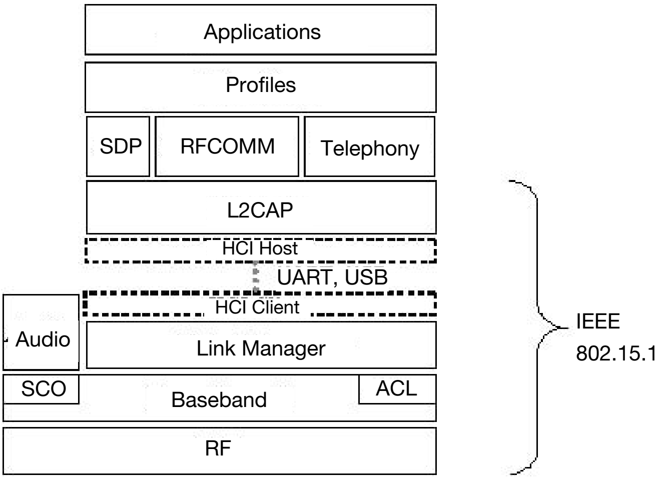


Figure 2.3. Simplified Bluetooth protocol stack.

quency Shit Keying). According to the transmitted power, Bluetooth devices can be classified into different power classes from 20 dBm to 0 dBm transmission power. Class-3 devices are the most common, transmitting with 0 dBm, and not requiring external power amplification or power control; thus, they can be integrated on a single chip.

2.3.2.2 The Baseband Layer. The Baseband layer is in charge of controlling the RF layer and providing the communications structure to the higher layers, thus taking on the functions of the MAC sublayer of the OSI-7. The basic communication structure provided by Bluetooth is a point-to-point link between two devices, each of them hopping along the same pseudorandom sequence of frequency carriers. In order for the two nodes to agree on the hopping sequence and on the control of the channel, one of the nodes will assume the role of master while the other becomes a slave. (Nodes do not have to be different in their capabilities, the master/slave roles are logical roles in the point-to-point communication link.) A point-to-multipoint *Piconet* can be established by a single master controlling the channel for several slaves (the point-to-point communication structure in general is also called a Piconet). A Piconet only has one master and can have several slaves hopping along the pseudorandom sequence of the master of the Piconet, with a maximum channel capacity of 1 Mbps shared by the members of the Piconet. As mentioned earlier, a functioning Piconet makes 1600 hops in a second, thus having 1600 slots (each 625 μs long) in one second. In an odd-numbered slot, only the master of the Piconet is allowed access (with a few exemptions); whereas, in an even-numbered slot, a slave that was polled in the previous slot can gain access to the channel. To enable Bluetooth devices to tune to the new frequency carrier and change their mode from reception to transmission, a 220 μs *guard time* is set aside at the end of transmission slots, thus reducing the goodput. Nodes are also enabled to transmit during not only one but three and five slots, using different packet types (with no hopping while in transmission) to increase efficiency by reducing the "effective usage time–guard time" ratio. The effective data rate in a Piconet can be determined according to the packet types and lies anywhere between 216 kbps and 780 kbps per Piconet. Several Piconets can operate in the same space independently without causing a significant interference among each other, since all these Piconets will hop according to different hopping sequences. The probability of interference between independent Piconets grows by the number of Piconets covering the same area. It is also worth noting that the 2.4 GHz band is also utilized by other (interfering) technologies such as IEEE802.11b and microwave ovens.

There are two different types of classifications for the virtual links between nodes in a Piconet: a link can be Synchronous Connection Oriented (SCO) or Asynchronous Connectionless (ACL). If an SCO link is established between two nodes of a Piconet, then slots are reserved at fixed intervals for the master and one of its slaves in the Piconet, ensuring a deterministic assignment of slots to the traffic. SCO links provide a voice-type quality of service provisioning, indeed designed for voice transmissions. ACL links, on the other hand, are in sole control of the master polling the slaves in the order the master desires. Slots assigned to SCO links have priority over ACL links as well as priority over any other task a master may be performing (e.g., inquiring or paging).

As mentioned earlier, the basic communication structure of Bluetooth is a Piconet; thus, Piconets need to be established over Bluetooth devices before they can exchange data or communicate. The Piconet establishment process is a three-step process including device discovery (or inquiry, in Bluetooth terms), device attachment (or paging, in Bluetooth terms), and Piconet parameter negotiations.

During the *inquiry* process, the common objective of Bluetooth nodes is to discover each other's presence with some of the nodes listening or *scanning* the (reduced set) of hopping frequencies while other nodes constantly transmit very short so-called ID packets. Since inquiry ID packets are extremely short and represent a unique bit pattern, the number of hops can be increased to 3200 hops per second to reduce discovery times. If a scanning node overhears an ID packet for the first time, it will refrain from replying immediately but will wait a random (back-off) period of time to reduce the collision probability of scanning nodes replying to the same ID packet. When finished with the backlogging, nodes return to the inquiry scan state, and, if they overhear another ID, packet they will respond to the transmitter of that ID packet in exactly 625 μs. The inquiring nodes send two ID packets at two different frequencies and then listen to the corresponding reply frequencies for the next 625 μs if reply is received. The inquiring node will be aware of the proximity and the identity of the scanning node.

The *paging* process can start if there are devices that are aware of the identities other devices in their proximity, most likely after a successful inquiry. Just like with the inquiry process, the frequency of the hopping is increased to 3200 and devices can be either in a page scan or page mode. By definition, the node that initiates the paging (the node in the page mode) will become the master of the Piconet, whereas the node that was successfully paged will become the slave. The device in the paging mode will transmit an ID packet with the address of the device it has discovered before. If the device whose ID is transmitted is in the page scan mode and overhears the ID packet with its own address, then it will respond to this "page" with the same ID packet. Note that the paging node knows the identity of the paged device but not necessarily vice versa; thus, the paging node that received a reply from the paged node will send an identification packet with its own parameters to the paged node (the latter responding with another ID packet). By the time this four-way handshake is executed, the slave (paged node) has enough information to calculate the master node's pseudorandom hopping sequence so both the nodes can start using the hopping sequence of the master, establishing a *connection.*

Reaching the connection state, the master will poll the slave to verify that the slave has entered the Piconet. The third phase of the connection establishment is initiated by the Link Manager layer to set up a control ACL link.

A Piconet can consist of a maximum of eight active nodes: a master and seven active slaves. This is due to three-bit node addressing inside Piconets. Yet, a Piconet can consist of much more devices in an inactive mode; indeed, the number of nonactive slave devices in a Piconet is not constrained. Other than being actively participating in a Piconet, slaves can go or be put into three different power saving modes: Sniff, Hold, and Park. A slave in Sniff mode will not listen to the channel in every odd time slot, but will negotiate a parameter with the master for periodic small time windows during which it will wake up and check whether the master wants to transmit to it. The Sniff mode can be used to reduce power consumption of rarely active nodes. In the Hold mode (just like in the Sniff mode), a slave still does not give up its three-bit active-address but will not be able to receive any ACL packets for a negotiated period of time. The Hold mode may be used to perform inquiry and scanning operations while being connected to a Piconet and to enable the participation of nodes in more than one Piconet, as outlined later. Finally, slaves in the Park mode give up the three-bit active address but will remain synchronized to the master by listening to the channel during so-called Beacon intervals. If a master wants to wake up a parked slave, it will have to wait for the negotiated Beacon window and address the slave to be awaked with the device address or parked address. Parked slaves will also receive an opportunity during the Beacon window to inform the master that they need to be woken up.

Although the main communication unit in Bluetooth is a point-to-multipoint Piconet, the specification allows nodes to participate in more than one Piconet semisimultaneously (note that a node can be a master in only one Piconet), switching between its roles of the different Piconets acting as bridges between Piconets, likely using the Hold mode to schedule between the several Piconets. Two or more overlapping Piconets interconnected with bridges in such manner form a *Scatternet.* Although a Piconet's topology is a star-shaped point-to-multipoint structure with only a single link between a master and any of its slaves (single-hop), a Scatternet can represent any type of the possible topologies and, thus, can be used to establish a multihop or *ad hoc network* (a possible Scatternet is depicted in Figure 2.4). Other than describing the possibility of forming Scatternets, the Bluetooth specification does not address how Scatternets or ad hoc networks should be established; it solely provides the possibility to employ Bluetooth as the basis for ad hoc networking.

2.3.2.3 Link Manager. The Link Manager (LM) layer of Bluetooth fulfils part of the functionality of the Logical Link Control sublayer of the OSI-7 architecture. The main functions of the LM are: Piconet management, link configuration, and providing security, that is, authentication and encryption. Right after a slave has been put into a Connection mode, an ACL link is established between master and slave to manage the Piconet. Management functions include the attachment and detachment of slaves, negotiating piconet parameters, a possible change in the roles (when a slave becomes the new master of the Piconet), the establishment of SCO or ACL links, and the handling of the low-power modes. The management functions are based on a request–response communication scheme between the master and the slave, whereby the master requests some parameter to be changed and the slave either accepts it or challenges it.

The link configuration tasks consist of (i) quality of service negotiations, whereby the maximum polling time is negotiated in a request–response manner and broadcast parameters are set up; (ii) negotiation of power-control parameters; (iii) negotiation of accepted packet types at both sides, with determination of whether multislot packets will be allowed.

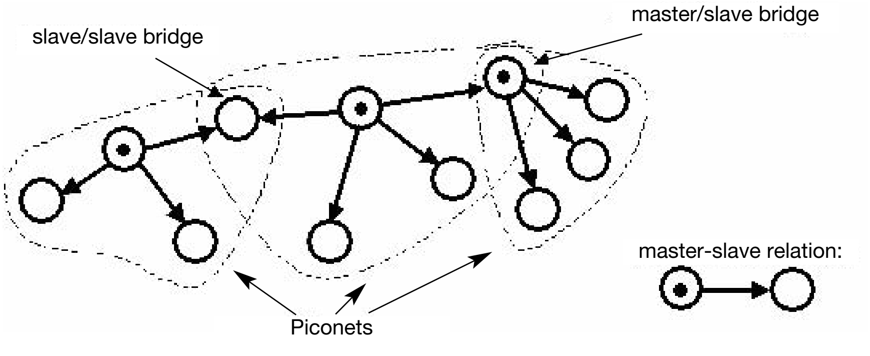


Figure 2.4. A Bluetooth Scatternet consisting of three Piconets.

The security goals include (i) optional authentication to only allow devices that are known or trusted to connect, and (ii) encryption to prevent eavesdropping by a third party on the channel. Authentication is based on a common link key, whereby the verifier challenges the claimant to compute an answer that can only be computed by knowing the link key. In order to distribute the link key, nodes go through a process called *pairing.* During pairing, a link key is formed from a PIN code, a random number, and the claimants address. For encryption of data, an encryption key length is negotiated between master and slave and an encryption key is created using the same algorithm at both sides and the link key.

2.3.2.4 Logical Link Control and Adaptation Protocol Layer. The Logical Link Control and Adaptation Protocol Layer (L2CAP) is the other subprotocol of the Logical Link Control sublayer of the OSI-7 protocol stack. The goals of L2CAP are to enable several higher-layer protocols to transmit their protocol data units (PDU) over ACL links (protocol multiplexing), segmentation and reassembly of higher layer PDUs into Baseband packets, and quality of service negotiations for individual ACL links for the higher-layer protocols. The L2CAP can provide both connection-oriented and connectionless communication to the higher layers. L2CAP is needed for protocol multiplexing, since the headers of Baseband packets does not include bits to specify what higher-layer protocol is encapsulated in the Baseband packet. The L2CAP protocol header contains logical channel identification bits with which connection-oriented protocol multiplexing can be done, whereas, for connection less services and control information fixed, special channel identifiers are used. Segmentation and reassembly takes care of using several of the small Baseband packets to transmit higher-layer packets of size of up to 64 kB. Once the transmission of a segmented packet starts on the ACL link, no other L2CAP ACL packets can be interleaved with the transmission; the transmitting of the whole higher-layer PDU has to be finished first. The L2CAP layer does not support SCO links nor does it perform integrity checks. It assumes that data integrity issues are taken care of at the Baseband layer with automatic retransmissions or forward-error corrections.

2.3.2.5 Higher Layers and Bluetooth Profiles. The Bluetooth specification defines higher-layer protocols, that is, protocols to emulate several serial connections over ACL links, such as the RFCOMM protocol, and a protocol that defines how devices can find out about what services other devices provide, such as the Service Discovery Protocol (SDP). In the second volume of the specification, called the *Profiles,* different services are standardized for Bluetooth links, such as headset profiles, serial port profiles, intercom profiles, LAN access profiles, file transfer profiles, and synchronization profiles. Bluetooth devices connected to each other can query the profiles the other device is offering and, if they implement the same profiles, they can establish connections for the given profiles, ensuring interoperability.

These protocols and profiles do not reside at the bottom two layers of the OSI-7 model and, thus, are not an integral part of IEEE802.15.1. Since they have little significance to ad hoc networks, their functional description is omitted in this chapter.

2.3.2.6 Further Reading. Readers interested in more details of the Bluetooth specification are referred to the freely available Bluetooth specifications [7, 8], to books summarizing the Bluetooth specifications, such as [10, 28], or to the many available white papers and general resources on Bluetooth on the World-Wide-Web.

2.3.3 Using Bluetooth for Ad Hoc Networking

Scatternet functionality is essential for Bluetooth to be used as an enabler for ad hoc networks; discussions on such possibilities were ongoing as early as the first appearance of the Bluetooth specification. Unfortunately, only few of the commercially available Bluetooth kits implement Scatternet functionality (most of them do not even implement the power saving modes or point-to-multipoint operations). In order to use Bluetooth as an ad hoc network enabler, several research problems have to be solved, including an efficient way to discover other devices (inquiry) [45], an efficient way to switch bridge nodes between Piconets (Scatternet scheduling) [29], efficient ways to schedule the polling in multislave Piconets (Piconet scheduling) [11], selecting the best links to be activated for a Piconet [27], and having distributed algorithms forming Scatterenets (Scatternet formation) [46]. An enormous amount of research is focused on each of these areas; the cited references to each of these research areas only show a single representative publication for the reader who wants to get more details.

Chapter 4 of this book presents, investigates, and compares proposed Bluetooth Scatternet formation algorithms in detail.

Although several research groups plan to establish Bluetooth-based ad hoc networks, currently, no working testbed is available for study, so Bluetooth ad hoc network study remains in the simulation domain. IBM research has made their Bluetooth NS-2 extension [21] open-source available and the next release is supposed to have Scatternet functionality for simulation evaluation of Bluetooth-based ad hoc networks.

2.3.4 HomeRF—SWAP

The HomeRF Working Group [20] was launched in 1998 by Compaq, Intel, Motorola, National Semiconductor, Proxim, and Siemens to establish an industry standard supporting wireless home networks. Although enjoying the support of several big industry players, HomeRF has never taken off due to the popularity of IEEE802.11b. HomeRF positioned itself in the niche market of domestic users, which is why it is listed in this chapter under WPANs. HomeRF provides QoS-provisioned services, for example, for voice calls, as well as packet-switched best-effort services at the 2.4 GHz ISM band, with rates similar to that of IEEE802.11b (from specification 2.0), using FH technology. HomeRF's FH PHY layer was designed to work around interfering sources in the home environment, such as microwave ovens, by monitoring the channels and banning those channels from its hopping scheme that have too much interference.

The MAC layer protocol of HomeRF is called SWAP (Shared Wireless Access Protocol), which provides TDMA services for isochronous data and two different priorities of IEEE802.11, like CSMA/CA service for asynchronous data.

Although HomeRF products are available in limited supply, and nothing contradicts using the technology for ad hoc networking, the popularity and inexpensiveness of Wi-Fi preempts HomeRF for use as an enabler for ad hoc networking testbeds. Additionally, HomeRF is not an open standard, making its acceptance even more difficult.

2.3.5 RFID

Radio Frequency Identification (RFID) is a technology for providing a low-bandwidth, extremely inexpensive scheme for small integrated devices to talk wirelessly to access points relaying their ID (along with some optional data). RFID is used mainly for inventory purposes to be able to automatically monitor large inventories that are individually tagged with RFID tags. RFID can use active or passive tags. Passive tags do not have an internal power source but need to be placed in an electromagnetic field to be activated and readable, whereas active tags are battery operated and have a longer range, but are more expensive. Unfortunately, RFID technology lacks strong standards; most of the products available represent someone's proprietary technology.

The authors of this chapter are unaware of any serious RFID-based ad hoc network proposals, testbeds, or simulations, although RFID can be an extremely inexpensive basis for large-scale, low-rate-sensor ad hoc networks. The reader interested in more details about RFID is referred to the online resources [37, 38].

2.4 CONCLUSION

This chapter introduced several WPAN and WLAN standards. In the WLAN are, the strongest competitor today is IEEE802.11b, also called Wi-Fi 2.4 GHz. Wi-Fi 5.2 GHz (IEEE802.11a) is quickly emerging, providing symbol rates comparable to that of Fast-Ethernet. Wi-Fi defines an ad hoc operational mode, which makes it the most common off-the-shelf enabler for ad hoc network testbeds.

Bluetooth is the strongest standard in the WPAN field. Although the Bluetooth specification allows for the establishment of ad hoc networks, referred to as Scatterenets, there are major challenges that need to be overcome for Bluetooth to be considered a strong contender as an off-the-shelf ad hoc networking enabler.

Although it has been shown that none of these technologies is perfect for ad hoc networks, they will remain the premier choices for establishing testbeds. Ad hoc research will have to wait for dedicated ad hoc PHY and MAC technology until a killer application is defined for wide, commercial use of ad hoc networks.

Chapters 3 and 4 further investigate the use of the IEEE802.11 and Bluetooth technologies, respectively, for ad hoc networking.

ACKNOWLEDGMENTS

Our sincere gratitude goes to Rózsa P. Záruba and Wook Choi, who helped in collecting and compiling the references for this chapter.

REFERENCES

1. Official Homepage of The IEEE802.11 Working Group for Wireless LANs, http://grouper.ieee.org/groups/802/11/.
2. IEEE 802, "Get IEEE 802," http://standards.ieee.org/getieee802/.
3. 802–11 Planet Online Resource, http://www.80211-planet.com/.
4. Acticom R&D, http://www.acticom.de/1357.html.
5. Ad Hoc Protocol Evaluation Testbed, http://apetestbed.sourceforge.net/.
6. W. A. Arbaugh, "An Inductive Chosen Plaintext Attack Against WEP/WEP2," *IEEE Document 802.11-01/230,* May 2001.

7. Bluetooth SIG, “Specification of the Bluetooth System—Core,” vol. 1, version 1.1, http://www.bluetooth.com/dev/specifications.asp, February 2001.
8. Bluetooth SIG, “Specification of the Bluetooth System—Profiles,” vol. 2, version 1.1, http://www.bluetooth.com/dev/specifications.asp, February 2001.
9. N. Borisov, I. Goldberg, and D. Wagner, “Intercepting Mobile Communications: The Insecurity of 802.11,” in *Proceedings of the Seventh Annual International Conference on Mobile Computing and Networking (MOBICOM2001),* pp. 180–189, Rome, Italy, July 2001.
10. J. Bray, C. F. Sturman, and J. Mendolia, *Bluetooth 1.1: Connect Without Cables,* 2nd ed., Prentice-Hall, 2001.
11. A. Capone, M. Gerla, and R. Kapoor, “Efficient Polling Schemes for Bluetooth Picocells,” in *Proceeding of the IEEE International Conference on Communications (ICC2001),* vol. 7, pp. 1990–1994, Helsinki, Finland, June 2001.
12. I. Chen, “Wireless Ad Hoc Messenger,” a Virginia Tech and Microsoft project, http://people.cs.vt.edu/~irchen/microsoft-grant/description.html.
13. J. C. Chen and J. M. Gilbert, “Measured Performance of 5GHz 802.11a Wireless LAN Systems,” Atheros Communications White Paper, http://www.atheros.com/pt, 2001.
14. T. Clausen, P. Jacquet, A. Laouiti, P. Minet, P. Mulethaler, A. Qayyum, and L. Viennot, “Optimized Link State Routing Protocol,” IETF DRAFT, draft-ietf-manet-olsr-02.txt, http://hipercom.inria.fr/olsr/, July 2002.
15. The DAWN project, http://ntrg.cs.tcd.ie/dawn.php.
16. ETSI—BRAN, “ETSI HIPERLAN 1 Standards,” http://www.etsi.org/frameset/home.htm?/technicalactiv/Hiperlan/hiperlan1.htm.
17. ETSI—BRAN, “ETSI HiperLAN 2 Standards,” http://www.etsi.org/frameset/home.htm?/technicalactiv/Hiperlan/hiperlan2.htm.
18. Global Mobile Information Systems Simulation Library (GloMoSim), http://pcl.cs.ucla.edu/projects/glomosim/.
19. J. Habetha and M. Nadler, “Concept of Wireless Centralized Multihop Ad Hoc Network,” in *Proceedings of the European Wireless Conference,* Dresden, September 2002.
20. HomeRF Working Group, http://www.homerf.org.
21. IBM Research, BlueHoc: Open-Source Bluetooth Simulator, http://www–124.ibm.com/developerworks/opensource/bluehoc/.
22. IBM Zurich Research Laboratory, “Wireless Infrared Multipoint Network—Alr,” http://www.zurich.ibm.com/cs/wireless/usermodel.html.
23. D. B. Johnson, “Routing in Ad Hoc Networks of Mobile Hosts,” in *Proceedings of the. ACM MOBICOM ‘94,* December 1994.
24. M. Johnsson, “HiperLAN/2—The Broadband Radio Transmission Technology Operating in the 5GHz Frequency Band,” White Paper in HiperLAN 2 Global Forum, http://www.hiperlan2.com/technology.asp, 1999.
25. P. Karn, “MACA—A New Channel Access Protocol for Wireless LANs,” in *Proceedings of the ARRL/CRRL Amateur Radio 9th Computer Networking Conference,* pp.134–140, 1990.
26. A. McCormick, “OFDM Tutorial,” http://oldeee.see.ed.ac.uk/~acmc/OFDMTut.html.
27. Gy. Miklós, A. Rácz, Z. Turányi, A. Valkó, and P. Johansson, “Performance Aspects of Bluetooth Scatternet Formation,” in poster section of *MobiHoc 2000,* Boston, MA, August 2002.
28. B. A. Miller and C. Bisdikian, *Bluetooth Revealed: The Insider’s Guide to an Open Specification for Global Wireless Communications,* Prentice-Hall, 2000.
29. V. B. Misic and J. Misic. “Performance of Bluetooth Bridges in Scatternets With Limited Service Scheduling,” *ACM/Kluwer Journal of Mobile Networks and Applications (MONET),* special issue on Advances in Research of Wireless Personal Area Networking and Bluetooth Enabled Networks, 2002.

30. Mobile Ad Hoc Network Testbed (MART), http://www.cs.hut.fi/~mart/index.html.
31. The MOMENT Ad Hoc Network Testbed Project, http://moment.cs.ucsb.edu/projects.html.
32. The Monarch Project, http://www.monarch.cs.rice.edu/.
33. The Network Simulator—NS-2, http://www.isi.edu/nsnam/ns/.
34. The OFDM Forum, http://www.ofdm-forum.com.
35. OPNET modeler, http://www.opnet.com.
36. C. E. Perkins and P. Bhagwat, "Highly Dynamic Destination Sequenced Distance Vector Routing (DSDV) for Mobile Computers," in *Proceeding of the ACM SIGCOMM '94,* vol. 24, no. 4, p. 234, October 1994.
37. RFID Technologies, http://www.aimglobal.org/technologies/rfid/.
38. *RFID Journal,* www.rfidjournal.com.
39. B. Sklar, "Rayleigh Fading Channels in Mobile Digital Communications Systems Part I: Characterization," *IEEE Communications Magazine,* pp. 90–100, July 1997.
40. B. Sklar, "Rayleigh Fading Channels in Mobile Digital Communications Systems Part II: Mitigation," *IEEE Communications Magazine,* 102–109, July 1997.
41. Ultra-wideband Networking Group, http://www.uwb.org.
42. The Wireless Network Testbed, http://www.ee.surrey.ac.uk/CCSR/Mobile/Projects/Testbed/.
43. S. Xu and T. Saadawi, "Does the IEEE802.11 MAC Protocol Work Well in Multihop Wireless Ad Hoc Networks?," *IEEE Communications Magazine, 39,* 6, 130–137, June 2001.
44. O-C. Yue, "Design Trade-Offs in Cellular/PCS Systems," *IEEE Communications Magazine,* 146–152, September 1996.
45. G. V. Záruba, "Accelerated Neighbor Discovery in Bluetooth Based Personal Area Networks," in *International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA'02),* Las Vegas, NV, June 2002.
46. G. V. Záruba, I. Chlamtac, and S. Basagni, "Bluetrees—Scatternet Formation to Enable Bluetooth-Based Ad Hoc Networks," in *Proceedings of the IEEE International Conference on Communications (ICC2001),* pp. 273–277, Helsinki, Finland, June, 2001.

CHAPTER 3

IEEE 802.11 AD HOC NETWORKS: PROTOCOLS, PERFORMANCE, AND OPEN ISSUES

GIUSEPPE ANASTASI, MARCO CONTI, and ENRICO GREGORI

3.1 INTRODUCTION

The previous chapter has presented the activities of the different task groups within the IEEE 802.11 project [14], and has highlighted that the IEEE 802.11 is currently the most mature technology for infrastructure-based wireless LANs (WLANs). The IEEE 802.11 standard defines two operational modes for WLANs: *infrastructure-based* and *infrastructureless* or *ad hoc*. Network interface cards can be set to work in either of these modes but not in both simultaneously. The infrastructure-based is the mode commonly used to construct the so-called Wi-Fi hotspots, i.e., to provide wireless access to the Internet. The drawbacks of an infrastructure-based WLAN are the costs associated with purchasing and installing the infrastructure. These costs may not be acceptable for dynamic environments in which people and/or vehicles need to be temporarily interconnected in areas without a preexisting communication infrastructure (e.g., intervehicular and disaster networks), or where the infrastructure cost is not justified (e.g., in-building networks, specific residential community networks, etc.). In these cases, a more efficient solution can be provided by the infrastructureless or ad hoc mode.

When operating in this mode, stations are said to form an Independent Basic Service Set (IBSS) or, more simply, an ad hoc network. Any station that is within the transmission range of any other, after a synchronization phase, can start communicating. No Access Point (AP) is required, but if one of the stations operating in the ad hoc mode also has a

Mobile Ad Hoc Networking. Edited by Basagni, Conti, Giordano, and Stojmenovic.
ISBN 0-471-37313-3 © 2004 Institute of Electrical and Electronics Engineers, Inc.

connection to the wired network, stations forming the ad hoc network have a wireless access to the Internet.

The IEEE 802.11 technology is a good platform to implement single-hop ad hoc networks because of its extreme simplicity. Single-hop means that stations must be within the same transmission radius (say, 100–200 meters) to be able to communicate. This limitation can be overcome by multihop ad hoc networking. This requires the addition of routing mechanisms at stations so that they can forward packets toward the intended destination, thus extending the range of the ad hoc network beyond the transmission radius of the source station. Routing solutions designed for wired networks (e.g., the Internet) are not suitable for the ad hoc environment, primarily due to the dynamic topology of ad hoc networks.

In a pure ad hoc networking environment, the users' mobile devices *are* the network and they must cooperatively provide the functionality that is usually provided by the network infrastructure (e.g., routers, switches, and servers). This approach requires that the users' density be high enough to guarantee the packet forwarding among the sender and the receiver. When the users' density is low, networking may become unfeasible.

Even though large-scale multihop ad hoc networks will not be available in the near future, on smaller scales, mobile ad hoc networks are starting to appear, thus extending the range of the IEEE 802.11 technology over multiple radio hops. Most of the existing IEEE 802.11-based ad hoc networks have been developed in the academic environment but, recently, even commercial solutions have been proposed (see, e.g., MeshNetworks[1] and SPANworks[2]).

Other than being a solution for pure ad hoc networking, the IEEE 802.11 ad hoc technology may also constitute an important and promising building block for solving the first-mile problem in hotspots. This aspect is related to the understanding of some basic radio frequency (RF) transmission principles. Specifically, the transmission range is limited since the RF energy disperses as the distance from the transmitter increases. In addition, even though WLANs operate in the unregulated spectrum (i.e., the users are not required to be licensed), the transmitter power is limited by the regulatory bodies (e.g., the FCC in the United States and ETSI in Europe). IEEE 802.11a and IEEE 802.11b can operate at several bit rates but, since the transmitter power is limited, the transmission range decreases when the data rate is increased.

It is expected that the bandwidth request in hotspots will increase very fast, thus requiring higher-speed access technologies. As explained in the previous chapter in this book, channel speeds for the IEEE 802.11 family continue to increase: 802.11a operates at 54 Mbps, and enhanced versions operating at speeds up to 108 Mbps are also under investigation. Such high-speed WLAN standards are expected to further increase the popularity of wireless access to the backbone infrastructure. On the other hand, increasing the transmission rate (while maintaining the same transmission power) produces a reduction in the coverage area of an AP. Specifically, at the 100 Mbps rate the coverage area will correspond to a radius of few meters around the AP. It does not seem to be a feasible solution to spread in a hotspot a large number of APs uniformly and closely spaced. A more feasible solution may be based on the use of a relatively low number of multirate APs, and the deployment of multihop wireless networks that provide access to the wired backbone

[1] http://www.meshnetworks.com.

[2] http://www.spanworks.com.

via multiple wireless hops. When the population in a hotspot is low, the AP can use low transmission rates, thus covering a large area. In this case, the users' devices can contact the AP directly (i.e., single-hop). When the hotspot population increases, the data rate is increased as well and, hence, some devices can no longer directly contact the AP but must be supported by other devices for forwarding their traffic toward the AP. By further increasing the data rate, more users can be accommodated in the hotspot but, at the same time, more hops may be necessary for user traffic to reach the AP.

Currently, the widespread use of IEEE 802.11 cards makes this technology the most interesting off-the-shelf enabler for ad hoc networks. However, the standardization efforts have concentrated on solutions for infrastructure-based WLANs, whereas little or no attention has been given to the ad hoc mode. Therefore, the aim of this chapter is triple: (i) an in-depth investigation of the ad hoc features of the IEEE 802.11 standard, (ii) an analysis of the performance of 802.11-based ad hoc networks, and (iii) an investigation of the major problems arising when using the 802.11 technology for ad hoc networks, and possible directions for enhancing this technology for a better support of the ad hoc networking paradigm.

The rest of the chapter is organized as follows. The next section briefly describes the architecture and protocols of IEEE 802.11 WLANs. The aim is to introduce the terminology and present the concepts that are relevant throughout the chapter. The interested reader can find details on the IEEE 802.11 protocols in the standard documents [15].

The characteristics of the wireless medium and the dynamic nature of ad hoc networks make IEEE 802.11 multihop networks fundamentally different from wired networks. Furthermore, the behavior of an ad hoc network that relies upon a carrier-sensing random-access protocol such as the IEEE 802.11 is further complicated by the presence of hidden stations, exposed stations, "capturing" phenomena [28, 29], and so on. The interaction between all these phenomena makes the behavior of IEEE 802.11 ad hoc networks very complex to predict. Recently, this has generated an extensive literature related to the performance analysis of the 802.11 MAC protocol in the ad hoc environment, which we survey in Section 3.3. Most of these studies have been done through simulation. To the best of our knowledge, only very few experimental analyses have been conducted. For this reason, in Section 3.4 we extend the 802.11 performance analysis with an extensive set of measurements that we have performed on a real testbed. These measurements were performed both in indoor and outdoor environments, and in the presence of different traffic types. For the sake of comparison with the previous studies, our analysis is mostly related to the basic IEEE 802.11 MAC protocol (i.e., we consider a data rate of 2 Mbps). However, some results related to IEEE 802.11b are also included in Section 3.5. In the same section, we present some problems (gray zones) that may occur by using IEEE 802.11b in multihop ad hoc networks. Finally, in Section 3.6 we discuss some possible extensions to the IEEE 802.11 MAC protocol to improve its performance in multihop ad hoc networks.

3.2 IEEE 802.11 ARCHITECTURE AND PROTOCOLS

In this section, we will focus on the IEEE 802.11 architecture and protocols as defined in the original standard [15], with particular attention to the MAC layer. Later, in Section 3.5, we will emphasize the differences between the 802.11b standard with respect to the original 802.11 standard.

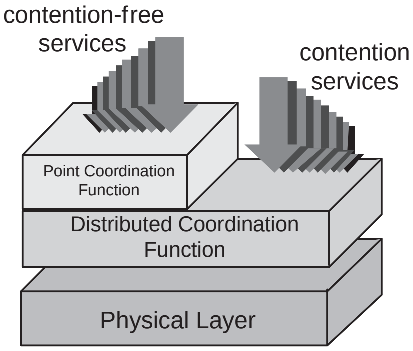

contention-free services
contention services
Point Coordination Function
Distributed Coordination Function
Physical Layer

Figure 3.1. IEEE 802.11 Architecture.

The IEEE 802.11 standard specifies both the MAC layer and the Physical Layer (see Figure 3.1). The MAC layer offers two different types of service: a contention-free service provided by *the Distributed Coordination Function* (DCF), and a contention-free service implemented by the *Point Coordination Function* (PCF). These service types are made available on top of a variety of physical layers. Specifically, three different technologies have been specified in the standard: Infrared (IF), Frequency Hopping Spread Spectrum (FHSS), and Direct Sequence Spread Spectrum (DSSS).

The DCF provides the basic access method of the 802.11 MAC protocol and is based on a *Carrier Sense Multiple Access with Collision Avoidance* (CSMA/CA) scheme. The PCF is implemented on top of the DCF and is based on a polling scheme. It uses a *Point Coordinator* that cyclically polls stations, giving them the opportunity to transmit. Since the PCF cannot be adopted in the ad hoc mode, it will not be considered hereafter.

3.2.1 Distributed Coordination Function (DCF)

According to the DCF, before transmitting a data frame, a station must sense the channel to determine whether any other station is transmitting. If the medium is found to be idle for an interval longer than the *Distributed InterFrame Space* (DIFS), the station continues with its transmission[3] (see Figure 3.2). On the other hand (i.e., if the medium is busy), the transmission is deferred until the end of the ongoing transmission. A random interval, henceforth referred to as the *backoff time,* is then selected; it is used to initialize the *backoff timer.* The backoff timer is decreased for as long as the channel is sensed as idle, stopped when a transmission is detected on the channel, and reactivated when the channel is sensed as idle again for more than a DIFS. (For example, the backoff timer of Station 2 in Figure 3.2 is disabled while Station 3 is transmitting its frame; the timer is reactivated a DIFS after Station 3 has completed its transmission.) The station is enabled to transmit its frame when the backoff timer reaches zero. The backoff time is slotted. Specifically, the backoff time is an integer number of slots uniformly chosen in the interval (0, *CW*-1). *CW* is defined as the Backoff Window, also referred to as the *Contention Window.* At the first transmission attempt, $CW = CW_{\min}$, and it is doubled at each retransmission up to $CW_{\max}$.

[3] To guarantee fair access to the shared medium, a station that has just transmitted a packet and has another packet ready for transmission must perform the backoff procedure before initiating the second transmission.

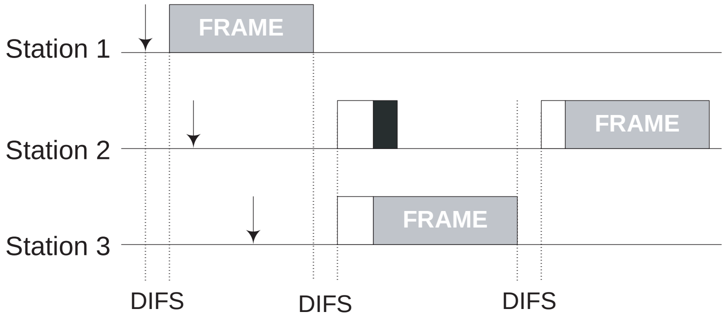


Figure 3.2. Basic access mechanism.

In the standard, $CW_{\min}$ and $CW_{\max}$ values depend on the physical layer adopted. For example, for the FHSS Phisical Layer $CW_{\min}$ and $CW_{\max}$ values are 16 and 1024, respectively [15].

Obviously, it may happen that two or more stations start transmitting simultaneously and a collision occurs. In the CSMA/CA scheme, stations are not able to detect a collision by hearing their own transmissions (as in the CSMA/CD protocol used in wired LANs). Therefore, an immediate positive acknowledgement scheme is employed to ascertain the successful reception of a frame. Specifically, upon reception of a data frame, the destination station initiates the transmission of an acknowledgement frame (ACK) after a time interval called the *Short InterFrame Space* (SIFS). The SIFS is shorter than the DIFS (see Figure 3.3) in order to give priority to the receiving station over other possible stations waiting for transmission. If the ACK is not received by the source station, the data frame is presumed to have been lost, and a retransmission is scheduled. The ACK is not transmitted if the received packet is corrupted. A *Cyclic Redundancy Check* (CRC) algorithm is used for error detection.

After an erroneous frame is detected (due to collisions or transmission errors), a station must remain idle for at least an *Extended InterFrame Space* (EIFS) interval before it reactivates the backoff algorithm. Specifically, the EIFS shall be used by the DCF whenever the physical layer has indicated to the MAC that a frame transmission was begun that did not result in the correct reception of a complete MAC frame with a correct FCS value.

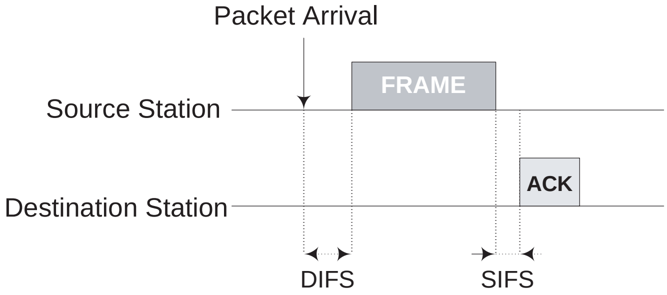


Figure 3.3. Interaction between the source and destination stations. The SIFS is shorter than the DIFS.

Reception of an error-free frame during the EIFS resynchronizes the station to the actual busy/idle state of the medium, so the EIFS is terminated and normal medium access (using DIFS and, if necessary, backoff) continues following reception of that frame.

3.2.2 Common Problems in Wireless Ad Hoc Networks

In this section, we discuss some problems that can arise in wireless networks, mainly in the ad hoc mode. The characteristics of the wireless medium make wireless networks fundamentally different from wired networks. Specifically, as indicated in [15]:

- The wireless medium has neither absolute nor readily observable boundaries outside of which stations are known to be unable to receive network frames.
- The channel is unprotected from outside signals.
- The wireless medium is significantly less reliable than wired media.
- The channel has time-varying and asymmetric propagation properties.

In a wireless (ad hoc) network that relies upon a carrier-sensing random-access protocol, like the IEEE 802.11 DCF protocol, the wireless medium characteristics generate complex phenomena such as the hidden-station and exposed-station problems.

Figure 3.4 shows a typical "hidden-station" scenario. Let us assume that station B is in the transmitting range of both A and C, but A and C cannot hear each other. Let us also assume that A is transmitting to B. If C has a frame to be transmitted to B, according to the DFC protocol, it senses the medium and finds it free because it is not able to hear A's transmissions. Therefore, it starts transmitting the frame but this transmission will result in a collision at the destination Station B.

The hidden-station problem can be alleviated by extending the DCF basic mechanism through a *virtual carrier sensing* mechanism (also referred to as a floor acquisition mechanism) that is based on two control frames: *Request To Send* (RTS) and *Clear To Send* (CTS). According to this mechanism, before transmitting a data frame, the station sends a short control frame, named RTS, to the receiving station announcing the upcoming frame transmission (see Figure 3.5). Upon receiving the RTS frame, the destination station

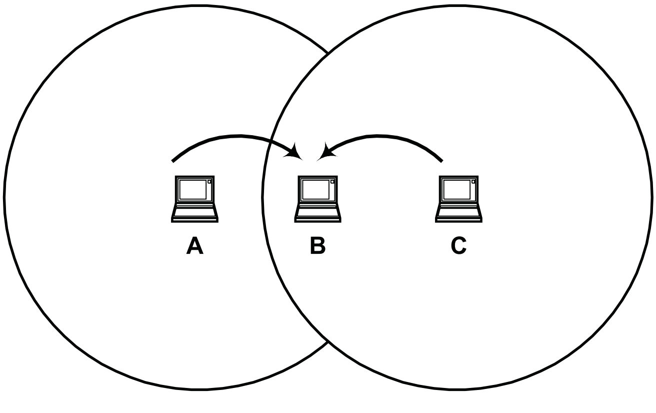


Figure 3.4. The "hidden-station" problem.

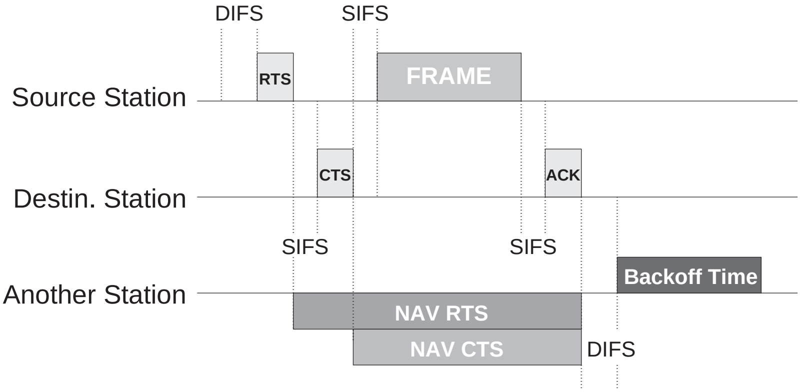


Figure 3.5. Virtual career sensing mechanism.

replies by sending a CTS frame to indicate that it is ready to receive the data frame. Both the RTS and CTS frames contain the total duration of the transmission, that is, the overall time interval needed to transmit the data frame and the related ACK. This information can be read by any listening station that uses this information to set up a timer called the *Network Allocation Vector* (NAV). When the NAV timer is greater than zero, the station must refrain from accessing the wireless medium. By using the RTS/CTS mechanism, stations may become aware of transmissions from hidden stations and learn how long the channel will be used for these transmissions.

Figure 3.6 depicts a typical scenario in which the "exposed-station" problem may occur. Let us assume that both Station A and Station C can hear transmissions from B, but Station A cannot hear transmissions from C. Let us also assume that Station B is transmitting to Station A and Station C receives a frame to be transmitted to D. According to the DCF protocol, C senses the medium and finds it busy because of B's transmission. Therefore, it refrains from transmitting to C, although this transmission would not cause a collision at A. The "exposed-station" problem may thus result in a throughput reduction.

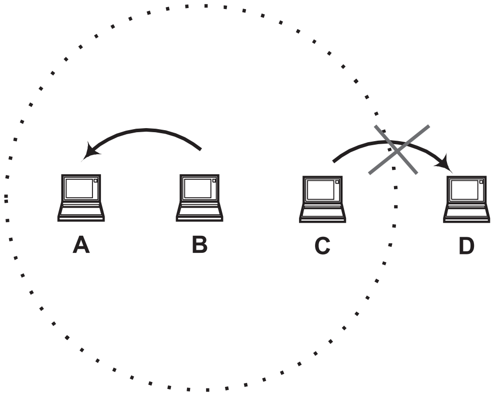


Figure 3.6. The "exposed-station" problem.

3.2.3 Ad Hoc Networking Support

In this section, we will describe how two or more 802.11 stations set up an ad hoc network. In the IEEE 802.11 standard, an ad hoc network is called an *Independent Basic Service Set* (IBSS). An IBSS enables two or more 802.11 stations to communicate each other without the intervention of either a centralized AP or an infrastructure network. Hence, the IBSS can be considered as the support provided by the 802.11 standard for mobile ad hoc networking.[4]

Due to the flexibility of the CSMA/CA protocol, to receive and transmit data correctly it is sufficient that all stations within the IBSS are synchronized to a common clock. The standard specifies a *Timing Synchronization Function* (TSF) to achieve clock synchronization between stations. In an infrastructured network, the clock synchronization is provided by the AP, and all stations synchronize their own clock to the AP's clock. In an IBSS, due to the lack a centralized station, clock synchronization is achieved through a distributed algorithm. In both cases, synchronization is obtained by transmitting special frames, called beacons, containing timing information.

The TSF requires two fundamental functionalities, namely *synchronization maintenance* and *synchronization acquirement,* that will be sketched below. We only focus on IBSS.

3.2.3.1 Synchronization Maintenance. Each station has a *TSF timer* (clock) with modulus 2^{64} counting in increments of microseconds. Stations expect to receive beacons at a nominal rate defined by the a *Beacon Period* parameter. This parameter is decided on by the station initiating the IBSS, and is then used by any other station joining the IBSS. Stations use their TSF timers to determine the beginning of beacon intervals or periods. At the beginning of a beacon interval, each station performs the following procedure:

1. It suspends the decrementing of the backoff timer for any pending (nonbeacon) transmission.
2. It generates a random-delay interval uniformly distributed in the range between zero and twice the minimum value of the Contention Window.
3. It waits for the random delay.
4. If a beacon arrives before the random delay timer has expired, it stops the random-delay timer, cancels the pending beacon transmission, and resumes the backoff timer;
5. If the random delay timer has expired and no beacon has been received, it sends a beacon frame.

The sending station sets the beacon timestamp to the value of its TSF timer at the time the beacon is transmitted. Upon reception of a beacon, the receiving station looks at the timestamp. If the beacon timestamp is later than the station's TSF timer, the TSF timer is set to the value of the received timestamp. In other words, all stations within the IBSS synchronize their TSF timer to the quickest TSF timer.

[4] To uniquely identify a IBSS it is necessary to associate to it an identification number (IBSSID) that is locally administered and that will be used by any other Station to join the IBSS, i.e., the ad hoc network. When a station starts a new IBSS, it generates a 46-bit random number in a manner that minimizes the probability that the same number is generated by another station.

3.2.3.2 Synchronization Acquirement. This functionality is necessary when a station wants to join an already existing IBSS. The discovery of existing IBSSs is the result of a scanning procedure of the wireless medium during which the station receiver is tuned to different radio frequencies, looking for particular control frames. Only if the scanning procedure does not result in finding any IBSS, the station may start with the creation of a new IBSS. The scanning procedure can be either passive or active.

In passive scanning, the station listens to the channels for a beacon frame. It is worth repeating that a beacon frame contains not only timing information for synchronization, but also the complete set of IBSS parameters. This set includes the IBSS identifier IBSSID, the aBeaconPeriod parameter, the data rates that can be supported, and the parameters relevant to IBSS management functions (e.g., power-saving management).

Active scanning involves the generation of Probe frames, and the subsequent processing of received Probe Response frames. The station that decides to start an active scanning procedure has a ChannelList of radio frequencies that will be scanned during the procedure. For each channel to be scanned, a probe with broadcast destination is sent by using the DCF access method. At the same time, a ProbeTimer is started. If no response to the probe is received before the ProbeTimer reaches the MinChannelTime, the next channel of the list is considered. Otherwise, the station continues to scan the same channel until the timer reaches the MaxChannelTime. Then, the station processes all received Probe responses.

Probe responses are sent using normal frame transmission rules as directed frames to the address of the station that generated the Probe request. In an IBSS, only the station that generated the last beacon transmission will respond to a probe request, in order to avoid wasting bandwidth with repetitive control frames. In each IBSS, at least one station must be awake at any given time to respond to the Probe request. Therefore, the station that sent the last beacon remains in the awake state in order to respond to Probe requests, until a new beacon is received. There may be more than one station in a IBSS that responds to a given probe request, particularly in the case where more than one station transmitted a beacon, either due to not successfully receiving a previous beacon, or due to collision between beacon transmissions.

3.2.4 Power Management

In a mobile environment, portable devices have limited energy resources since they are powered by batteries. Power-management functionalities are thus extremely important both in the infrastructure-based and in the ad hoc modes. Obviously, in the ad hoc mode, that is, inside an IBSS, power-saving (PS) strategies need to be completely distributed in order to preserve the self-organizing nature of the IBSS. A station may be in one of two different power states: *awake* (the station is fully powered) or *doze* (the station is not able to transmit or receive). Multicast and/or directed frames destined to a power-conserving station are first announced during a period when all stations are awake. An Ad Hoc Traffic Indication Map (ATIM) frame does the announcement. A station operating in the PS mode listens to these announcements and, based on them, decides whether it has to remain awake or not.

ATIM frames are transmitted during the ATIM Window, a specific period of time following the beginning of a beacon period whose length is defined by the aATIMWindow parameter (an IBSS parameter included in the beacon content). During the ATIM Window, only beacon and ATIM frames can be exchanged and all stations must remain awake.

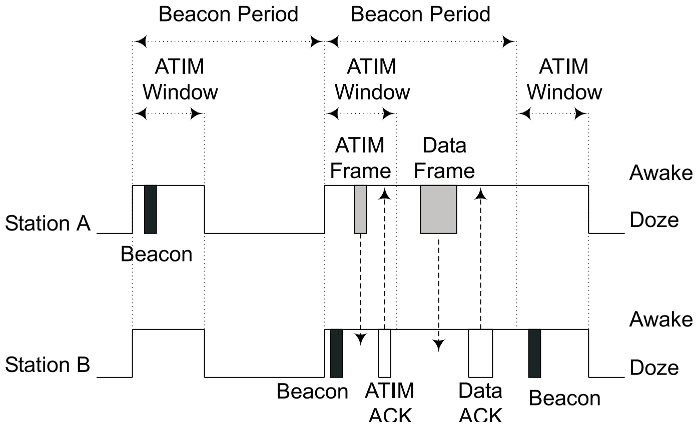


Figure 3.7. A data exchange between stations operating in PS mode in an ad hoc network.

Directed ATIM frames are to be acknowledged by the destination station, whereas multicast ATIMs are not to be acknowledged. Hence, a station sends a directed ATIM frame and waits for the acknowledgment. If this acknowledgement does not arrive, it executes the backoff procedure for retransmitting the ATIM frame.

A station receiving a directed ATIM frame must send the acknowledgement and remain awake for the entire duration of the beacon interval, waiting for the announced data frame. Data frames are transmitted at the end of the ATIM Window according to the DCF access method (see Figure 3.7). If a station does not receive any ATIM frame during the ATIM Window, it can enter the doze state at the end of the ATIM window.

3.3 SIMULATION ANALYSIS OF IEEE 802.11 AD HOC NETWORKS

As mentioned above, in this chapter we are primarily interested in the performance provided by the 802.11 MAC protocol in an ad hoc environment. In this framework, almost all previous works are based on simulation and have looked at the performance of TCP applications. Less attention has been devoted to UDP applications (this can be easily justified since, currently, the most popular applications use TCP as the transport protocol).

The previous studies have been pointed out several performance problems. They can be summarized as follows. In a dynamic environment, mobility may have a severe impact on the performance of the TCP protocol [2, 4, 6, 9, 12, 13, 17]. However, even when stations are static, the performance of an ad hoc network may be quite far from ideal. It is highly influenced by the operating conditions, that is, TCP parameter values (primarily the congestion window size) and network topology [9, 16]. In addition, the interaction of the 802.11 MAC protocol (hidden- and exposed-station problems, exponential backoff scheme, etc.) with TCP mechanisms (congestion control and time-out) may lead to unexpected phenomena in a multihop environment. For example, in the case of simultaneous TCP flows, severe unfairness problems and, in extreme cases, capture of the channel by few flows [24, 26–29] may occur. Even in the case of a single TCP connection, the instantaneous throughput may be very unstable [26, 27]. Such phenomena do not appear, or appear with less intensity, when the UDP protocol is used [29].

In the next subsections, we will briefly survey the findings of the previous studies. To better understand the results presented below, it is useful to provide a model of the relationships existing among stations when they transmit or receive. In particular, it is useful to make a distinction between the transmission range, the interference range, and the carrier sensing range. The following definitions can be given.

The *Transmission Range* (TX_range) is the range (with respect to the transmitting station) within which a transmitted packet can be successfully received. The transmission range is mainly determined by the transmission power and the radio propagation properties.

The *Physical Carrier Sensing Range* (PCS_range) is the range (with respect to the transmitting station) within which the other stations detect a transmission. It mainly depends on the sensitivity of the receiver (the receive threshold) and the radio propagation properties.

The *Interference Range* (IF_range) is the range within which stations in the receive mode will be "interfered with" by a transmitter, and thus suffer a loss. The interference range is usually larger than the transmission range, and it is a function of the distance between the sender and receiver, and of the path loss model. It is very difficult to predict the interference range as it strongly depends on the ratio between power of the received "correct" signal and the power of the received "interfering" signal. Both these quantities heavily depend on several factors (i.e., distance, path, etc.) and, hence, to estimate the interference we must have a detailed snapshot of the current transmission and relative station position.

In the simulation studies presented hereafter, the following relationship has been generally assumed: *TX_range ≤ IF_range ≤ PCS_range.* For example, in the ns-2 simulation tool [20] the following values are used to model the characteristics of the physical layer: *TX_range* = 250 m, *IF_range = PCS_range* = 550 m.

3.3.1 Influence of Mobility

Station mobility may severely degrade the performance of the TCP protocol in mobile ad hoc networks (MANETs) [2, 4, 6, 9, 12, 13, 17]. This is due to the inability of the TCP protocol to manage efficiently the effects of mobility. Station movements may cause route failures and route changes and, hence, packet losses and delayed ACKs. The TCP misinterprets these events as congestion signals and activates the congestion control mechanism. This leads to unnecessary retransmissions and throughput degradation. In addition, mobility may exacerbate the unfairness between competitive TCP sessions [24].

Numerous new mechanisms have been proposed for optimizing the TCP performance in MANETs, including the adaptation of TCP error-detection and recovery mechanisms to the mobile ad hoc environment. Chandran et al. propose to introduce explicit signaling (Route Failure and Route Re-establishment notifications) from intermediate stations to notify the sender TCP of the disruption of the current route, and construction of a new one [4]. Upon receiving a route failure notification, the sender TCP does not activate the congestion control mechanism, but simply freezes its status, which will be resumed when a Route Establishment notification has been received.

In [13] an Explicit Link Failure Notification (ELFN) is still used to notify the sender TCP about a route failure. However, no explicit signaling about route reconstruction is provided. Monks, Sinha, and Bharghavan present a simulation study of the ELFN mechanism, both in static and dynamic scenarios [19]. This study points out the limitations of this approach that are intrinsic to TCP properties (e.g., long recovery time after a timeout), and proposes to implement mechanisms below the TCP layer. A similar approach is taken in [17] where the standard TCP is unmodified but new mechanisms are implemented in a thin layer, Ad hoc TCP (ATCP), between TCP and IP. ATCP uses Explicit Congestion Notifications (ECN) and ICMP "destination unreachable" messages to discriminate congestion conditions from link failures, and from packet losses in wireless links. The ATCP takes the appropriate actions according to the type of event recognized.

All previous techniques require an explicit notification from intermediate stations to the sender TCP. To avoid this complexity, a heuristic is used in [6] to distinguish route failures from congestions. When timeouts occur consecutively, the sender TCP assumes that a route failure occurred rather than a network congestion. The unacknowledged packet is retransmitted again but the retransmission timeout is not doubled a second time. The retransmission timeout remains fixed until the route is reestablished and the packet is acknowledged. An implicit detection approach is also taken in [31], where the authors propose to infer route changes by observing the out-of-order delivery events.

3.3.2 Influence of the Network Topology

Even in a static environment, the performance of an ad hoc network is strongly limited by the interaction between neighboring stations [16]. Stations' activity is limited by the behavior of neighboring stations (a station must sense the medium before it starts transmitting) and by stations in its interfering range (interferences may cause collisions at the destination station). For example, it can be shown that in a string (or chain) topology, like the one shown in Figure 3.8, the expected maximum bandwidth utilization is only 0.25 [16]. However, things may be even worse in practice. This discrepancy is due to 802.11 MAC inability to find the optimum schedule of transmissions by itself. In particular, in a chain topology it happens that stations early in the chain starve later stations (similar considerations apply to other network topologies). In general, the 802.11 MAC protocol appears to be more efficient in the case of local traffic patterns, that is, when the destination is close to the sender [16].

3.3.3 Influence of the TCP Congestion Window Size

The TCP congestion window size may have a significant impact on performance. In [10], it is shown that, for a given network topology and traffic pattern, there exists an optimal value of the TCP congestion window size at which the channel utilization is maximized. However, the TCP does not operate around this optimal value and typically grows its average window size much larger, leading to decreased throughput (throughput degradation is in the order of 10–30% with respect to the optimal case) and increased packet losses. This behavior can be explained by considering the origin of packet losses, which in ad hoc networks is completely different from that in traditional wired networks. In ad hoc networks,

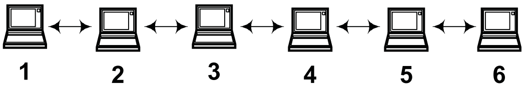


Figure 3.8. A string network topology.

packet losses caused by buffer overflows at intermediate stations are rare events (unless the station buffer is very small), whereas packet losses due to link-layer contention (i.e., a station that fails to reach its adjacent station; see Section 3.3.4 and [26]) are largely dominant. Furthermore, the multihop wireless network collectively exhibits graceful loss behavior. In general, the link loss probability is insufficient to stabilize the average TCP congestion window around the optimal value. To achieve this objective, Fu and others propose two link-level mechanisms: Link RED, and adaptive spacing [10]. Similarly to the RED mechanism implemented in Internet routers, the Link RED tunes the packet loss probability at the link layer by marking/discarding packets according to the average number of retries experienced in the transmission of previous packets. The Link RED thus provides TCP with an early sign of overload at the link level. Adaptive spacing is introduced to improve spatial channel reuse, thus reducing the risk of stations' starvation. The idea here is the introduction of extra backoff intervals to mitigate the exposed receiver problems. Adaptive spacing is complementary to Link RED: it is activated only when the average number of retries experienced in previous transmission is below a given threshold.

3.3.4 Effects of the Interaction between MAC Protocol and TCP Mechanisms

The interaction of some features of the 802.11 MAC protocol (hidden-/exposed-station problem, exponential backoff scheme, etc.) with the TCP protocol mechanisms (mainly, the congestion control mechanism) may lead to several unexpected and serious problems. S. Xu and Saadawi identified these problems through a simulation analysis of a multihop ad hoc network via the *ns* network simulator tool [26]. The same results have been confirmed with a different simulation tool [27]. Recently, similar phenomena have been also observed in other scenarios [27, 28].

Specifically, in [26] and [27] it is pointed that the following problems may affect the TCP performance in a multihop ad hoc environment:

1. The instantaneous throughput of a TCP connection may be very unstable (dropping frequently to zero), even when this is the only active connection in the network (*instability problem*).
2. In case of two simultaneous TCP connections, it may happen that the two connections can not coexist: when one connection develops, the other one is shut down (*incompatibility problem*).
3. With two simultaneous TCP connections, if one connection is single-hop and the other one is multiple-hop, it may happen that the instantaneous throughput of the multiple-hop connection is shut down as soon as the other connection becomes active (even if the multiple-hop connection starts first). There is no chance for the multiple-hop connection once the one-hop connection has started (*one-hop unfairness problem*).

The above problems have been revealed in a string network topology like the one shown in Figure 3.8, where the distance between any two neighboring stations is 200 m and stations are static. According to the 802.11-based Wave-Lan, the nominal transmission radius of each station has been set to 250 m (each station can thus communicate only with its neighboring stations). Furthermore, the sensing and interfering ranges have been set to twice the transmission range of 500 m [26, 27], which is the typical setting of the ns-2 simulator.

Below we will provide a brief explanation of how the one-hop unfairness problem arises. Similar explanations can be provided for the instability and incompatibility problems, but are omitted for the sake of space. The reader can refer to [27] for a detailed analysis of all cases.

Figure 3.9 shows two TCP connections. The first connection is from Station 2 to Station 3 (one-hop connection), whereas the second connection is from Station 6 to Station 4 (two-hop connection). Let us assume, for example, that Station 2 is transmitting a data frame to Station 3 (e.g., a TCP segment), and Station 5 wants to transmit a frame to Station 4. According to the 802.11 MAC protocol, Station 5 tries to send an RTS frame, and then waits for the corresponding CTS frame. However, Station 5 never receives this CTS frame.

Most of the RTS transmission attempts tried by Station 5 result in a collision at Station 4 due to the interference of Station 2 (hidden-station problem). Station 5 cannot hear the CTS from Station 3 because it is out of the transmission range of Station 3 and, thus, it is not aware of Station 2 transmission. However, Station 4 is in the interfering range of Station 2 since the interfering range is larger than the transmission range (twice in ns-2 simulator). Even if Station 4 successfully receives the RTS frame, it is not able to reply with the corresponding CTS frame, again due to Station 2. Though Station 4 is out of the transmission range of Station 2, Station 4 can sense the transmission of Station 2 since the sensing range is larger than the transmission range (twice in the ns-2 simulator). This inhibits Station 4 from accessing the wireless medium (exposed-station problem).

After failing to receive the CTS frame from Station 4 seven times, Station 5 reports a link breakage to its upper layer and a route-failure notification is sent to Station 6 (the data packet originator). Upon receiving this notification, Station 6 starts a route discovery process. Obviously, while looking for a new route no data packet can flow along the connection and this makes the instantaneous throughput drop to zero.

The above example allows us to understand why the instantaneous throughput of the two-hop connection drops to zero. However, it not yet clear why this throughput remains at zero for most of the connection lifetime. To better clarify this point, the following additional remarks need to be taken into account.

- Since Station 5 is in the interfering range of Station 3, it has to defer when Station 3 is sending. Therefore, Station 5 can transmit an RTS frame only when Station 3 is not sending.

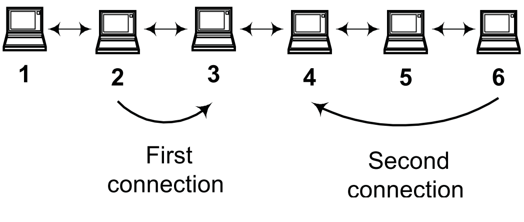


Figure 3.9. A string topology with two TCP connections. The first connection is one-hop; the second connection is two-hop.

- In the one-hop connection, as soon as Station 2 receives a TCP ACK from Station 3, it immediately prepares itself to send another TCP segment. This means that Station 5 has very few opportunities to find the channel free.
- Data frames (i.e., TCP segments) transmitted by Station 2 are usually much larger in size than the RTS frames that Station 5 tries to transmit.

In conclusion, the time available for Station 5 for successfully accessing the channel is very small. In addition, the exponential backoff scheme used by the 802.11 MAC protocol always favors the last succeeding station.

From the above description, it emerges that several features of the multihop ad hoc environment contribute to the "capture" of the channel by the one-hop connection. The most important and direct causes are the hidden-station and the exposed-station problems. These problems, in turn, are caused by the larger size of the interfering and sensing ranges with respect to the transmission range. However, the random backoff scheme of the 802.11 MAC protocol also contributes by favoring the last succeeding station.

The "capture" effect revealed in [26, 27] is not peculiar to the string network topology. Gerla et al. observed the same phenomenon even in other scenarios [28, 29]. In [28], they also proposed two possible solutions to remove the capture effect: (i) replacement of the binary backoff scheme in the 802.11 MAC protocol by an adaptive retransmission timeout based on the number of active neighboring stations; and (ii) the use of special antennas that reduce interference during packet reception.

3.4 EXPERIMENTAL ANALYSIS OF IEEE 802.11 AD HOC NETWORKS

In the previous section, we have seen that there exists an extensive literature that has investigated TCP performance in ad hoc networks, especially over the IEEE 802.11 MAC protocol. Most papers report the same type of unfairness problems. The hidden- and exposed-station problem, the large interference range, and the backoff scheme of IEEE 802.11 MAC protocol have been recognized as the major reasons for these unfairness problems. All these previous analyses were carried by simulation and, hence, the results observed are highly dependent on the physical layer model implemented in the simulation tool used in the analysis (e.g., GloMosim [11], ns-2 [20], Qualnet [23]). Hereafter, we validate and extend these results by presenting a similar analysis that has been carried on a real testbed. Since the simulation results presented in Section 3 were obtained by considering IEEE 802.11 network cards operating at the nominal bit rate of 2 Mbps, most of the measurement studies presented in this section refer to the IEEE 802.11 standard [15]. However, in Section 3.5 we will also investigate the performance of the IEEE 802.11b ad hoc networks.

It is worth pointing out that, although in the simulation studies presented above the values of *TX_range, PCS_range,* and *IF_range* are known and constant, in the real world the physical channel has time-varying and asymmetric propagation properties. Hence, the values of *TX_range, PCS_range,* and *IF_range* may be highly variable, even during the same experiment.

3.4.1 Experimental Testbed

The measurement testbed is based on laptops running the Linux-Mandrake 7.2 operating system. The laptops are equipped with Lucent WaveLAN IEEE 802.11 network cards using the DSSS technique, and operating at the nominal bit rate of 2 Mbps. The target of our study is the analysis of the TCP performance over an IEEE 802.11 ad hoc network. Since the aim of the study is to investigate the impact of the CSMA/CA protocol on the TCP performance, static ad hoc networks (i.e., the network stations do not change their position during an experiment) with single-hop TCP connections were considered. This allows one to remove other possible causes that may interfere with the TCP behavior, such as, link breakage, route recomputation, etc.

3.4.2 Indoor Experiments

The indoor experiments were carried out in the scenario depicted in Figure 3.10. Stations numbered as S1, S2, and S3 have an active ftp session toward Station S4; that is, data frames are transmitted to S4, which replies with ACK packets. As ftp data transfers are supported by the TCP protocol, in the following the data flows will be denoted as TCP*i*, where *i* is the index of the transmitting station. As shown in the figure, a reinforced concrete wall (represented by the gray rectangle) is located between stations S1 and S2, and between stations S2 and S3. As a consequence, S1, S2, and S3 are outside the *TX_ range* of each other.[5] Furthermore, each Station S*i* (where $i = \{1, 2, 3\}$) is in the transmission range of S4. Therefore, this is a typical hidden-station scenario in which it is expected that the RTS/CTS mechanism (by avoiding hidden-station collisions) should provide a significant throughput gain with respect to the basic CSMA/CA protocol.

Two sets of experiments were performed in this scenario. In the first set, only two ftp sessions are active: TCP1 and TCP2. In the second set, all three sessions are active.

To better analyze the results, we also performed some reference experiments. Specifically, we measured the maximum throughput (at the application layer) of a single sender–receiver session when the two stations are very close to each other (in the same room), and no other session is active. The estimated throughput represents the upper bound throughput for a sender–receiver session and is reported in Table 3.1 for different operating conditions.

Let us now start analyzing the results related to the indoor scenario. The results obtained in the scenario with two active sessions (TCP1 and TCP2) are summarized in Figure 3.11. These results refer to a 60 second ftp transfer that utilizes TCP packets with a 1460 byte payload size. Two types of experiments were done: with and without the RTS/CTS mechanism. For each type, we performed three experiments under the same conditions.

The following remarks can be made based on the above results:

1. The RTS/CTS mechanism does not provide any significant performance improvement with respect to the basic access mechanism.
2. The RTS/CTS mechanism provides an aggregate throughput slightly lower than the basic access mechanism. This is due to the additional overhead introduced by the RTS and CTS frames.
3. In each experiment, the aggregate throughput is not very far from the reference throughput reported in Table 3.1 (i.e., 145 and 135 Kbps for the basic access and the RTS/CTS mechanism, respectively).

[5] This was verified by running the Ping program for a sufficiently long time from each station to the other stations. In no case was a packet successfully delivered among each couple of stations.

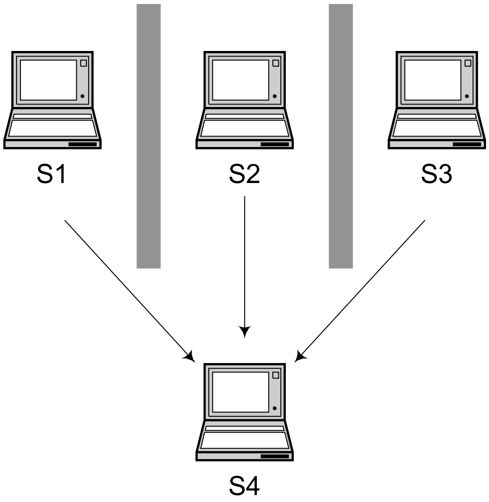


Figure 3.10. Indoor scenario.

These observations are confirmed by the results obtained in the scenario with three ftp sessions active, summarized in Table 3.2. For each set of experiments, Table 3.2 reports the throughput averaged on all the experiments performed under the same conditions.

These results indicate that the carrier sensing mechanism is still effective even if the transmitting stations are "apparently" hidden from each other. This can be explained by remembering that the carrier sensing range is about twice the transmission range. Hence, if two stations (outside the transmission range of each other) are in the transmission range of a third station, there is a very high probability that they can sense each other. In these cases, the physical carrier sensing is effective and, hence, adding virtual carrier sensing (i.e., the RTS/CTS mechanism) is useless.

3.4.3 Outdoor Experiments

To better investigate the phenomena observed in the indoor environment, the testbed was moved to an outdoor space. Each station was located in an open environment (a field without buildings) in order to analyze the TCP behavior when hidden and/or exposed stations may be present. In all experiments, the WLAN was set to 2 Mbps.

The network scenario for the outdoor experiments is shown in Figure 3.12. In this scenario, we may have two contemporary active sessions. Specifically, Station S1 communi-

Table 3.1. Reference Throughputs in Kbytes/sec (Kbps)

	Packet size 1460 Bytes	Packet size 512 Bytes	
	ftp/TCP traffic	ftp/TCP traffic	CBR/UDP traffic
Throughputs Basic Access	145 Kbps	125 Kbps	165 Kbps
Throughputs RTS/CTS	135 Kbps	110 Kbps	140 Kbps

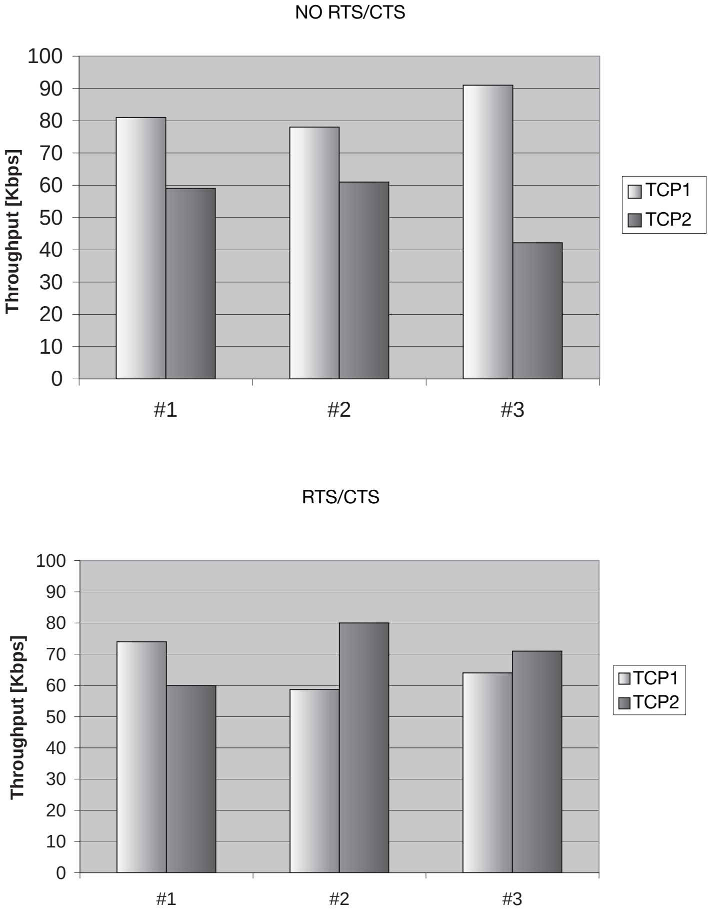


Figure 3.11. Throughput (in Kbps) estimated in the indoor scenario with two ftp sessions with the basic CSMA/CA access (left) and the RTS/CTS mechanism (right), respectively.

cates with Station S2 (Session 1), while Station S3 is in communication with Station S4 (Session 2). In the figure, the arrows represent the direction of the data flow (e.g., S1 is delivering data to S2), and $d(i, j)$ is the distance between stations Si and Sj. Data to be delivered are generated by either an ftp application, or a continuous bit rate (CBR) application. In the former, case the TCP protocol is used at the transport layer, whereas in the latter case UDP is the transport protocol.

Table 3.2. Throughput (in Kbps) Estimated in the Indoor Scenario when all Three ftp Sessions are Active

	TCP1	TCP2	TCP3	Aggregate
Basic Access	42	29.5	57	128.5
RTS/CTS	34	27	48	109

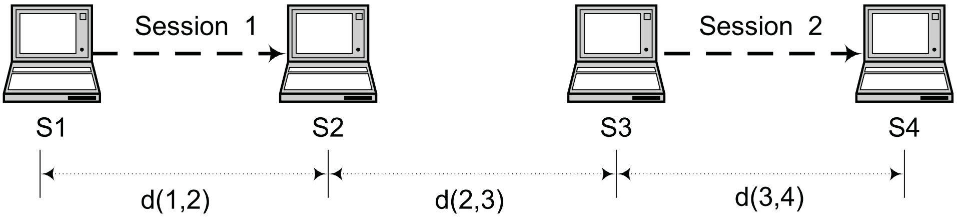


Figure 3.12. Reference network scenario for the outdoor experiments.

We performed a preliminary set of experiments aimed at estimating the *Tx_range* in the outdoor environment where the experiments were done. We used the following procedure. We considered a single couple of stations, S1 and S2. Then, starting from zero, we progressively increased the distance $d(1, 2)$, between these two stations until they were no longer able to exchange data. For each value of $d(1,2)$, the ping application was used to test the connectivity between the stations. By applying this procedure several times, we obtained that the transmission range is on the order of 40 m. It is worth pointing out that in a real environment, the value of *TX_range* is not constant. It is highly variable depending on several factors: weather conditions, hour of the day, place and time of the experiment, and so on.

Then we performed several experiments with Session 1 and Session 2 simultaneously active. In all experiments, the receiving station is always in the transmission range of its transmitting station; that is, Station S2 (S4) is in the transmitting range of Station S1 (S3). On the other hand, the distance $d(2, 3)$ between the two couples of stations[6] is variable. Depending on the actual $d(2, 3)$ value, the following situation can occur.

1. All stations are within the transmission range of each other (Type 1). This means that in our testbed, the distance between any two stations must be less than 40 m.
2. Extreme case: the two sessions are far from each other (Type 2). In our testbed, this is achieved by setting $d(2, 3) > 90$ m (i.e., more than twice the minimum transmission range size);.
3. Intermediate case 1 is obtained by setting $d(2, 3) = 65$ m (Type 3).
4. Intermediate case 2 is obtained by setting $d(2, 3) = 15$ m (Type 4).

In all experiments ftp data traffic was transmitted and the TCP protocol was used at the transport layer.[7] For this reason, the two sessions will be indicated below as TCP1 and TCP2. The payload size of TCP packets was set to 512 bytes.

The results obtained for Type 1 and Type 2 experiments are summarized in Table 3.3. These experiments produced the expected results. In Type 1 experiments (all stations within the same transmission range), the two ftp sessions fairly share the bandwidth, and the aggregate throughput is close to the reference throughput for this configuration (see Table 3.1). From the above results, it also appears that the RTS/CTS mechanism is useless since it only reduces the aggregate throughput (due to the overhead introduced by the RTS and CTS frames).

[6] That is, the couple (3, 4) with respect to the couple (1, 2), and vice versa.

[7] The length of each experiment is 120 seconds.

Table 3.3. Throughputs in Kbytes/sec (Kbps) Measured in Type 1 and Type 2 Experiments

	Type 1		Type 2	
	TCP 1	TCP 2	TCP 1	TCP 2
No RTS/CTS	61	54	122.5	122
RTS/CTS	59.5	49.5	96	100

In Type 2 measurements, the two sessions are independent, and they both achieve a throughput very close to the reference throughput. Again, the RTS/CTS mechanism is useless since it only introduces overhead.

Unlike the previous ones, Type 3 and Type 4 experiments exhibited a very strange and unpredictable behavior, as shown in Figure 3.13 and Figure 3.14. In Type 3 experiments, stations S2 and S3 are 65 m apart from each other. It can be observed that the use of the RTS/CTS mechanism produces a capture of the channel by the second session (i.e., S3–S4). A possible explanation for this behavior is that Station S2 is often blocked by S3 data transmissions to S4. Hence, it may not be able to reply to the RTS frame of S1. On the other hand, session S3–S4 is only marginally affected by session S1–S2, as the only possible impact is due to S3 being blocked by S2's (CTS and ACK) transmissions. When using the basic access mechanism, S1 can start transmitting to S2 without almost any interference from session S3–S4.

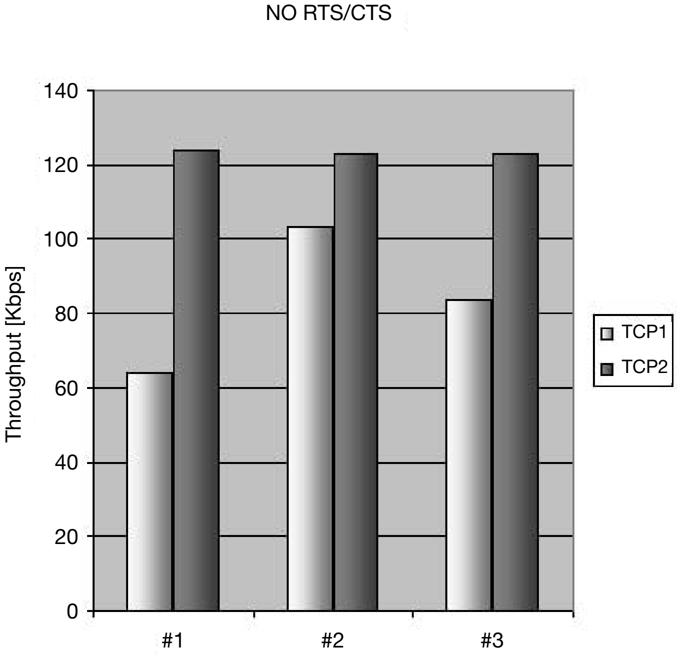
It is also worth noting that by using the basic access, the second session does not reduce its throughput (actually, the throughput of TCP2 increases as the RTS/CTS overhead is removed). Indeed, with the basic access each session achieves a higher throughput.

To summarize, in this configuration the RTS/CTS mechanism, adding further correlations between the stations' behavior (S1 cannot start transmitting if S2 does not reply with a CTS frame), produces a block of the first session without providing any advantage to the other one.

In Type 4 experiments, whose results are shown in Figure 3.14, we observed the capture of the channel by one of the two TCP connections. In this case, the RTS/CTS mechanism provided a little help in solving the problem.

The experimental results presented above confirm the unfairness/capture problems of the TCP protocol in IEEE 802.11 ad hoc networks revealed in previous simulation studies. As briefly discussed in Section 3.3, the TCP protocol (specifically the flow/congestion control mechanism), by introducing correlations in the transmitted traffic, emphasizes these phenomena. This effect is clearly pointed out by the experimental results shown in Figure 3.15. This figure still refers to the Type 4 configuration but traffic flows are now generated by CBR sources and the UDP protocol is used instead of TCP. As is clearly shown, the capture effects disappear.

In conclusion, the experimental results have confirmed that, in some scenarios, TCP connections may actually experience significant throughput unfairness, and even capture of the channel by one of the connections, as pointed out in previous simulation studies. Furthermore, it has been clearly shown that the RTS/CTS mechanism might be completely ineffective when there are stations that are outside their respective transmission ranges but within the same carrier sensing range. In these cases, the physical carrier sensing is sufficient to regulate the channel access and the virtual carrier sensing (i.e., the RTS/CTS mechanism) is useless.



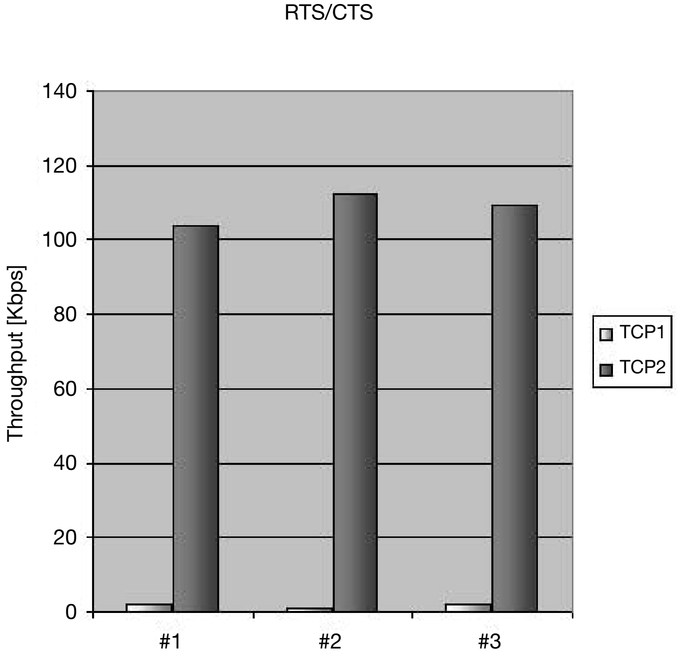


Figure 3.13. Throughputs (in Kbps) measured in the outdoor scenario in Type 3 experiments with the RTS/CTS mechanism disabled (top) and enabled (bottom).

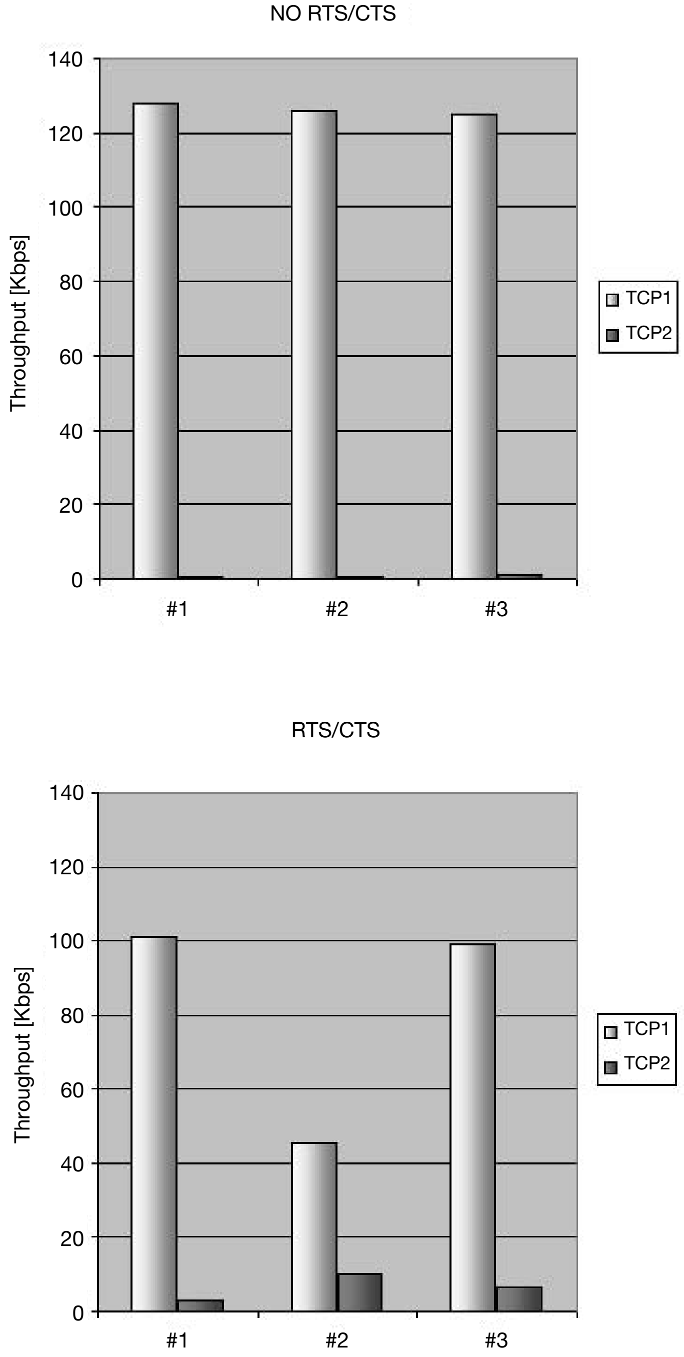


Figure 3.14. Throughputs (in Kbps) measured in the outdoor scenario in Type 4 experiments with the RTS/CTS mechanism disabled (top) and enabled (bottom).

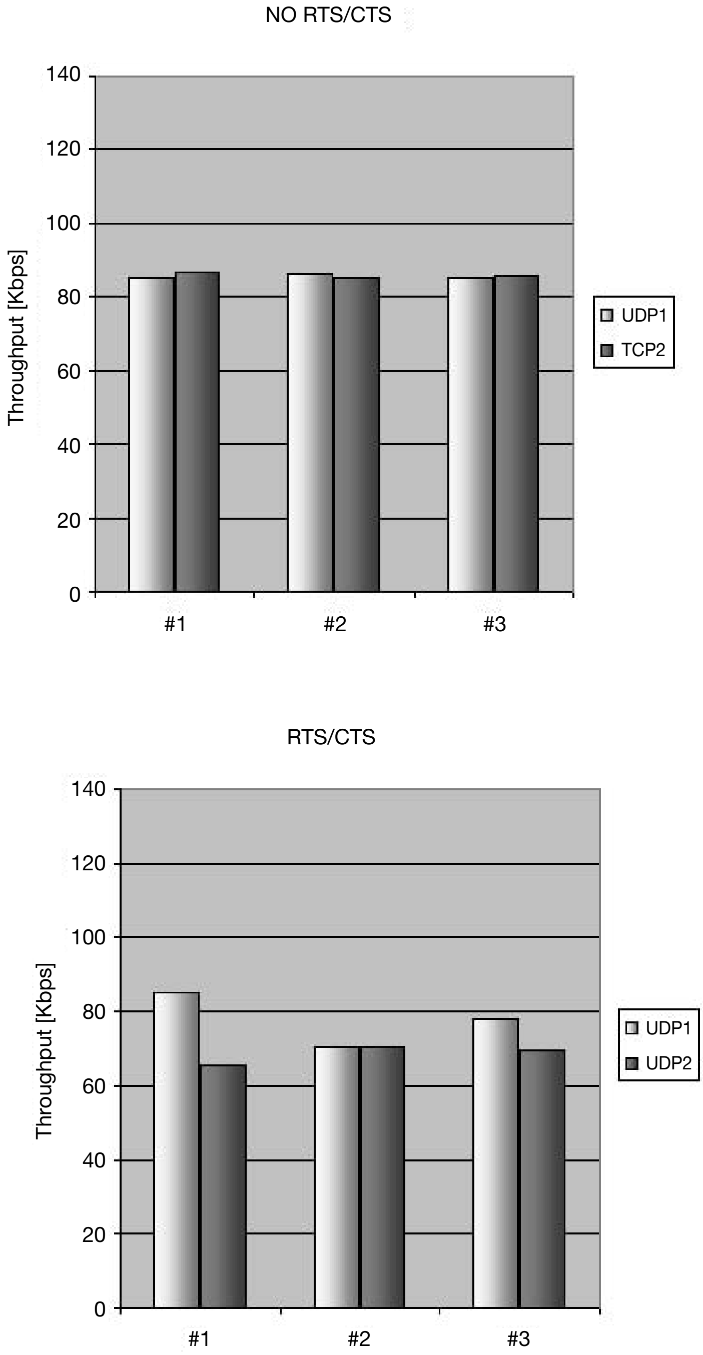


Figure 3.15. Type 4 experiments with CBR/UDP traffic.

3.5 IEEE 802.11b

The results presented in the previous section have been obtained by considering IEEE 802.11-based ad hoc networks. Currently, however, the Wi-Fi network interfaces are becoming more and more popular. Wi-Fi cards implement the IEEE 802.11b standard. It is therefore important to extend the previous experimental analysis to IEEE 802.11b ad hoc networks.

The 802.11b standard extends the 802.11 standard by introducing a higher-speed Physical Layer in the 2.4 GHz frequency band while still guaranteeing the interoperability with 802.11 cards. Specifically, 802.11b enables transmissions at 5.5 Mbps and 11 Mbps, in addition to 1 Mbps and 2 Mbps. 802.11b cards may implement a dynamic rate switching with the objective of improving performance. To ensure coexistence and interoperability among multirate-capable stations, and with 802.11 cards, the standard defines a set of rules that must be followed by all stations in a WLAN. Specifically, for each WLAN is defined a *basic rate set* that contains the data transfer rates that all stations within the WLAN must be capable of using to receive and transmit.

To support the proper operation of a WLAN, all stations must be able to detect control frames. Hence, RTS, CTS, and ACK frames must be transmitted at a rate included in the basic rate set. In addition, frames with multicast or broadcast destination addresses must be transmitted at a rate belonging to the basic rate set. These differences in the rates used for transmitting (unicast) data and control frames has a big impact on the system behavior, as clearly pointed out in [7].

Actually, since 802.11 cards transmit at a constant power, lowering the transmission rate permits the packaging of more energy per symbol, and this increases the transmission range. In the next subsections we investigate, by means of experimental measurements, (i) the relationship between the transmission rate of the wireless network interface card (NIC) and the maximum bandwidth utilization, and (ii) the relationship between the transmission range and the transmission rate.

3.5.1 Available Bandwidth

In this section, we will show that only a fraction of the 11 Mbps nominal bandwidth of IEEE 802.11b cards can be used for data transmission. To this end, we need to carefully analyze the overheads associated with the transmission of each packet (see Figure 3.16). Specifically, each stream of m bytes generated by a legacy Internet application is encapsulated by the TCP/UDP and IP protocols that add their own headers before delivering the resulting IP datagram to the MAC layer for the transmission over the wireless medium. Each MAC data frame is made up of (i) a *MAC header,* say MAC_{hdr}, containing MAC addresses and control information[8] and (ii) a variable-length *data payload,* containing the upper-layer data information. Finally, to support the physical procedures of transmission (carrier sense and reception), a *physical layer preamble* (PLCP preamble) and a *physical layer header* (PLCP header) have to be added to both data and control frames. Hereafter, we will refer to the sum of the PLCP preamble and PLCP header as PHY_{hdr}.

It is worth noting that these different headers and data fields are transmitted at different data rates to ensure the interoperability between 802.11 and 802.11b cards. Specifically,

[8] Without any loss of generality, we have considered the *frame error sequence* (FCS) for error detection as belonging to the MAC header.

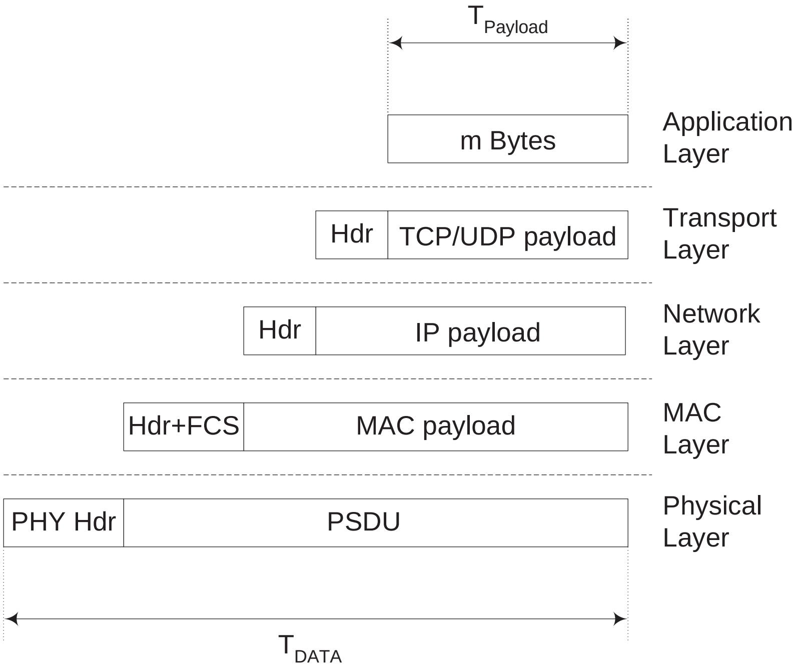


Figure 3.16. Encapsulation overheads.

the standard defines two different formats for the PLCP: Long PLCP and Short PLCP. Hereafter, we assume a Long PLCP that includes a 144 bit preamble and a 48 bit header, both transmitted at 1 Mbps, whereas the MAC_{hdr} and the MAC_{payload} can be transmitted at one of the NIC data rates: 1, 2, 5.5, and 11 Mbps. In particular, control frames (RTS, CTS, and ACK) can be transmitted at 1 or 2 Mbps, whereas the data frame can be transmitted at any of the NIC data rates.

By taking into consideration the above quantities, Equation 3.1 defines the maximum expected throughput for a single active session (i.e., only a sender–receiver couple is active) when the basic access scheme (i.e., DCF without RTS/CTS) is used. Specifically, Equation 3.1 is the ratio between the time required to transmit the user data and the overall time the channel is busy due to this transmission:

$$Th_{\text{noRTS/CTS}} = \frac{m}{DIFS + T_{\text{DATA}} + SIFS + T_{\text{ACK}} + \dfrac{CW\,\text{min}}{2} \cdot Slot\_Time} \qquad (3.1)$$

where
T_{DATA} is the time required to transmit a MAC data frame; this includes the PHY_{hdr}, MAC_{hdr}, MAC_{payload}, and FCS bits for error detection.
T_{ACK} is the time required to transmit a MAC ACK frame; this includes the PHY_{hdr} and MAC_{hdr}.
$\dfrac{CW\,\text{min}}{2} \cdot Slot\_Time$ is the average backoff time.

When the RTS/CTS mechanism is used, the overheads associated with the transmission of the RTS and CTS frames must be added to the denominator of Equation 3.1. Hence, in this case, the maximum throughput, $Th_{\mathrm{RTS/CTS}}$, is defined as

$$Th_{\mathrm{RTS/CTS}} = \frac{m}{DIFS + T_{\mathrm{RTS}} + T_{\mathrm{CTS}} + T_{\mathrm{DATA}} + T_{\mathrm{ACK}} + 3 \cdot SIFS + \dfrac{CW\,\mathrm{min}}{2} \cdot Slot\_Time} \tag{3.2}$$

where T_{RTS} and T_{CTS} indicate the time required to transmit the RTS and CTS frames, respectively.

The numerical results presented below depend on the specific setting of the IEEE 802.11b protocol parameters. Table 3.4 gives the values for the protocol parameters used hereafter.

In Table 3.5, we report the expected throughputs (with and without the RTS/CTS mechanism) by assuming that the NIC is transmitting at a constant data rate equal to 1, 2, 5.5., or 11 Mbps. These results are computed by applying Equations 3.1 and 3.2, and assuming a data packet size at the application level equal to $m = 512$ and $m = 1024$ bytes.

As shown in Table 3.5, only a small percentage of the 11 Mbps nominal bandwidth can be really used for data transmission. This percentage increases with the payload size. However, even with a large packet size (e.g., $m = 1024$ bytes) the bandwidth utilization is lower than 44%.

The above theoretical analysis has been complemented with the measurements of the actual throughput achieved at the application level. Specifically, we have considered CBR applications that exploit UDP as the transport protocol. Applications operate in asymptotic conditions (i.e., they always have packets ready for transmission) with constant-size packets of 512 bytes.

In Figure 3.17, the results obtained from this experimental analysis are compared with

Table 3.4. Value of the IEEE 802.11b Parameters

Slot_Time	τ	PHY_{hdr}	MAC_{hdr}	FCS	Bit Rate(Mbps)
20 μsec	≤1 μsec	192 bits (2.56 t_{slot})	240 bits (2.4 t_{slot})	32 bits (0.32 t_{slot})	1, 2, 5.5, 11
DIFS	*SIFS*	*ACK*		CSW_{MIN}	CSW_{MAX}
50 μsec	10 μsec	112 bits + PHY_{hdr}		32 t_{slot}	1024 t_{slot}

Table 3.5. Maximum Throughput at Different Data Rates

	m = 512 Bytes		*m* = 1024 Bytes	
	No RTS/CTS	RTS/CTS	No RTS/CTS	RTS/CTS
11 Mbps	3.337 Mbps	2.739 Mbps	5.120 Mbps	4.386 Mbps
5.5 Mbps	2.490 Mbps	2.141 Mbps	3.428 Mbps	3.082 Mbps
2 Mbps	1.319 Mbps	1.214 Mbps	1.589 Mbps	1.511 Mbps
1 Mbps	0.758 Mbps	0.738 Mbps	0.862 Mbps	0.839 Mbps

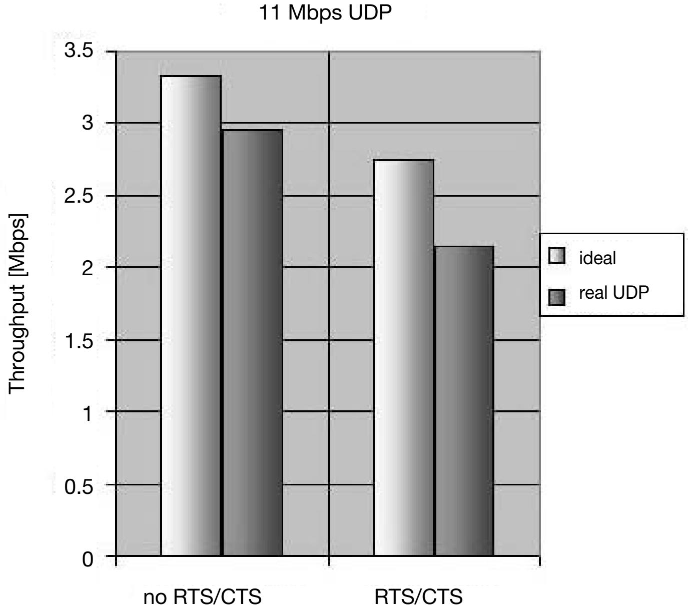


Figure 3.17. Comparison between the theoretical and the measured throughput.

the maximum expected throughputs calculated according to Equations 3.1 and 3.2. The real throughput is very close to the maximum throughput computed analytically. Similar results have been obtained by comparing the maximum throughput according to Equations 3.1 and 3.2 when the data rate is 1, 2, or 5.5 Mbps, and the real throughputs measured when the NIC bit rate is set accordingly.

3.5.2 Transmission Ranges

The dependency between the data rate and the transmission range was investigated by measuring the packet loss rate experienced by two communicating stations whose network interfaces transmit at a constant (preset) data rate. Specifically, four sets of measurements were performed corresponding to the different data rates: 1, 2, 5.5, and 11 Mbps. In each set of experiments, the packet loss rate was recorded as a function of the distance between the communicating stations. The resulting curves are presented in Figure 3.18.

Figure 3.19 shows the transmission-range curves derived on two different days (the data rate is equal to 1 Mbps). This graph highlights the variability of the transmission range depending on the weather conditions.

The results presented in Figure 3.18 are summarized in Table 3.6, where the estimates of the transmission ranges at different data rates are reported. These estimates point out that, when using the highest bit rate for data transmission, there is a significant difference in the transmission range of control and data frames. For example, assuming that the RTS/CTS mechanism is active, if a station transmits a frame at 11 Mbps to another station within its transmission range (i.e., less then 30 m apart) it reserves the channel for a radius of approximately 90 (120) m around itself. The RTS frame is transmitted at 2 Mbps (or 1 Mbps), and, hence, it is correctly received by all stations within the transmitting station's range, that is, 90 (120) meters.

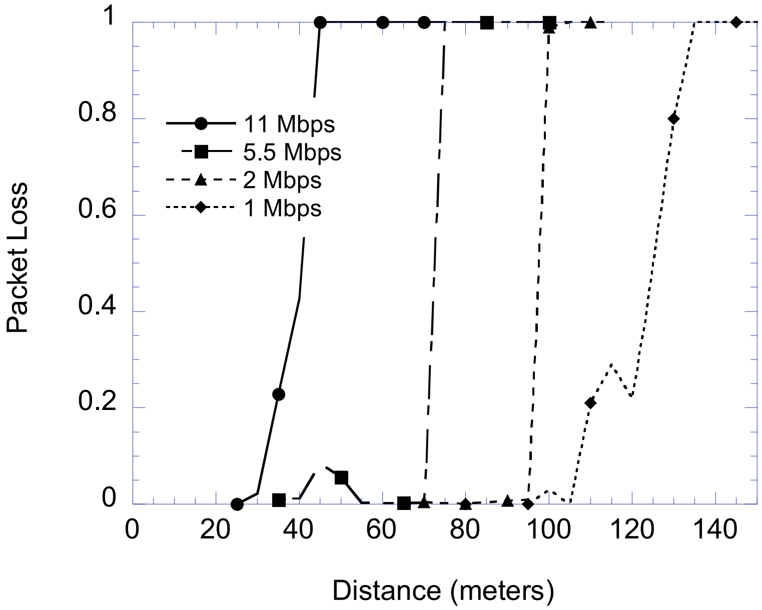


Figure 3.18. Packet loss rate as a function of the distance between communicating stations for different data rates.

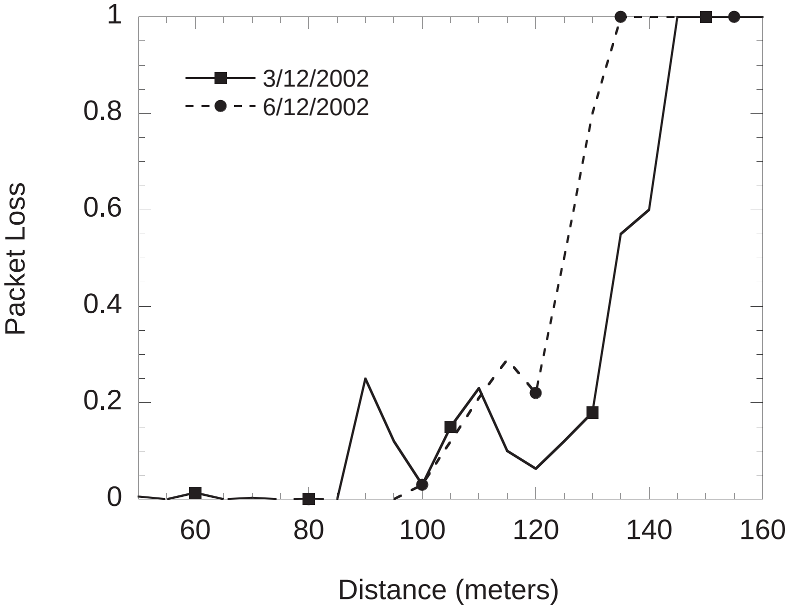


Figure 3.19. 1 Mbps transmission ranges on different days.

Table 3.6. Estimates of the Transmission Ranges at Different Data Rates

	11 Mbps	5.5 Mbps	2 Mbps	1 Mbps
Data *TX_range*	30 meters	70 meters	90–100 meters	110–130 meters
Control *TX_range*			≈ 90 meters	≈ 120 meters

Again, it is interesting to compare the transmission range used in the most popular simulation tools, like ns-2 and Glomosim, with the transmission ranges measured in our experiments. In these simulation tools it is assumed *TX_range* = 250 m. Since the above simulation tools only consider a 2 Mbps bit rate, we make reference to the transmission range estimated with a NIC data rate of 2 Mbps. As is clearly shown, the value used in the simulation tools (and, hence, in the simulation studies based on them) is two to three times higher that the values measured in practice. This difference is very important, for example, when studying the behavior of routing protocols: the shorter is the *TX_range*, the higher is the frequency of route recalculation when the network stations are mobile.

3.5.2.1 Transmission Ranges and the Mobile Devices' Height. During the experiments we performed to analyze the transmission ranges at various data rates, we observed a dependence of the transmission ranges on the mobile devices' height from the ground. Specifically, in some cases we observed that although the devices were not able to communicate when located on stools, they started to exchange packets when they were lifted up. In this section, we present the results obtained by a careful investigation of this phenomenon. Specifically, we studied the dependency of the transmission ranges on the devices' height from the ground. To this end we measured the throughput between two stations[9] as a function of their height from the ground. Four different heights were considered: 0.40 m, 0.80 m, 1.2 m, and 1.6 m. The experiments were performed with the Wi-Fi card set at two different transmission rates: 2 and 11 Mbps. In each set of experiments, the distance between the communicating devices was set in such a way as to guarantee that the receiver was always inside the sender's transmission range. Specifically, the sender–receiver distance was equal to 30 and 70 meters when the cards operated at 11 and 2 Mbps, respectively.

As clearly shown in Figure 3.20, the height may have a big impact on the quality of the communication between the mobile devices. For example, at 11 Mbps, by lifting up the devices from 0.40 meters to 0.80 meters, the throughput doubles, whereas further increasing the height does not produce significant throughput gains. Similar behavior is observed with a 2 Mbps transmission rate. However, in this case, the major throughput gain is obtained by lifting up the devices from 0.80 meters to 1.20 meters. A possible explanation for this different behavior is related to the distance between the communicating devices that is different in the two cases. This intuition is confirmed by the work presented in [21], which provides a theoretical framework to explain the height impact on IEEE 802.11 channel quality. Specifically, the channel power loss depends on the contact between the Fresnel zone and the ground. The Fresnel zone for a radio beam is an elliptical area with foci located in the sender and the receiver. Objects in the Fresnel zone cause diffraction and, hence, reduce the signal energy. In particular, most of the radio-wave energy is within the first Fresnel zone, the inner 60% of the Fresnel zone. Hence, if this inner part con-

[9] In these experiments, UDP is used as the transport protocol.

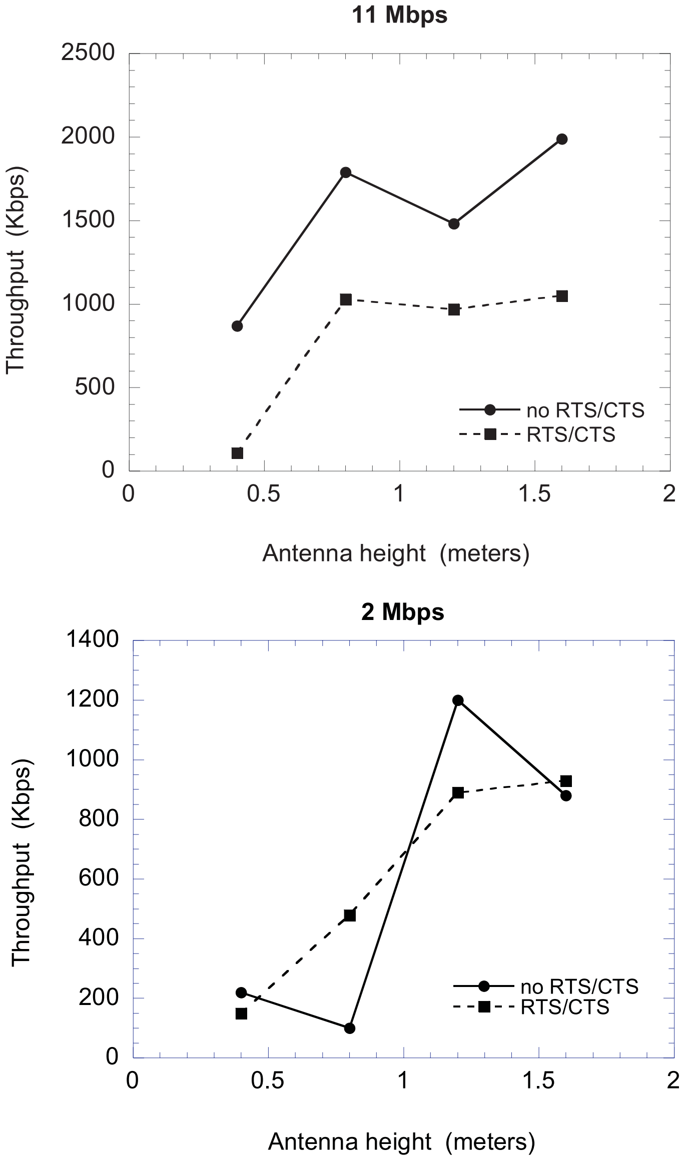


Figure 3.20. Relationship between throughput and devices' height.

tacts the ground (or other objects), the energy loss is significant. Figure 3.21 shows the Fresnel zone (and its inner 60%) for a sender–receiver couple at a distance D. In the figure, R1 denotes the height of the first Fresnel zone. As shown in [21], R1 is highly dependent on the station's distance. For example, when the sender and the receiver are at an height of 1 meter from the ground, the first Fresnel zone has contact with the ground only if $D > 33$ meters, whereas at heights of 1.5 and 2 meters, the first Fresnel zone contacts the ground only if D is greater than 73 and 131 meters, respectively. These theoretical computations are in line with our experimental results.

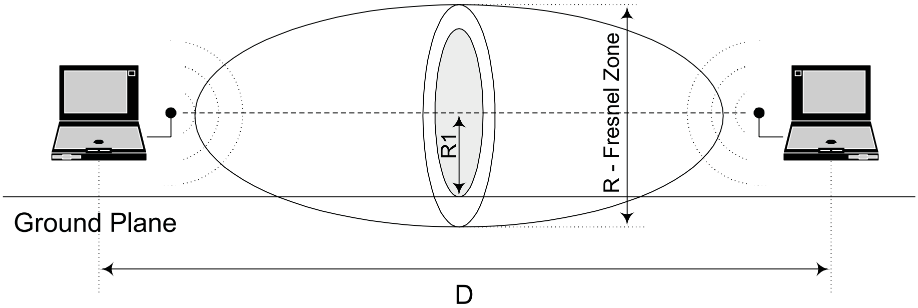


Figure 3.21. The Fresnel zone.

3.5.3 Four-Stations Network Configurations

The results presented in the previous sections show that the IEEE 802.11b behavior is more complex than the behavior of the IEEE 802.11 standard. Indeed, the availability of different transmission rates may cause the presence of several transmission ranges inside the network. In particular, inside the same data transfer session there may be different transmission ranges for data and control frame (e.g., RTS, CTS, ACK). Hereafter, we show that the superposition of these different phenomena makes it very difficult to understand the behavior of IEEE 802.11b ad hoc networks. To reduce this complexity, in the experiments presented below the NIC data rate is set to a constant value for the entire duration of the experiment.[10] Hereafter, we present only the results obtained with the NIC data rate constant and equal to 11 Mbps; more results can be found in [3].

The four-stations configuration presented in Figure 3.22 was used in the experiments. The results obtained are presented in Figure 3.23.

These results were the superposition of several factors. In detail, dependencies were observed between the two connections, even though the transmission range was smaller than the distance between stations S1 and S3. Furthermore, the dependency were observed also when the basic mechanism (i.e., no RTS/CTS) was used.[11] To summarize, this set of experiments showed that interdependencies among the stations extends beyond the transmission range. To explain this, we hypothesized that all stations were inside the same physical carrier sensing range, and this produced a correlation between active connections whose effect is similar to that achieved with the RTS/CTS mechanism (virtual carrier sensing). The difference in the throughputs achieved by the two sessions when using the UDP protocol (with or without RTS/CTS) can be explained by considering the asymmetric condition that exists on the channel: station S2 was exposed to transmissions of station S3 and hence, when station S1 sent a frame to S2, this station was not able to send back the MAC ACK. Therefore, S1 reacted as in the collision cases (thus rescheduling the transmission with a larger backoff). It is worth pointing out that also S3 was exposed to S2 transmissions but the S2's effect on S3 was less marked given the different role of the two stations. When using the basic access mechanism, the S2's effect on S3 was limited to short intervals (i.e., the transmission of ACK frames). When adopting the RTS/CTS

[10] It is worth pointing out that we experienced a high variability in the channel conditions, thus making a comparison between the results difficult.

[11] A similar behavior is observed (but with different values) by adopting the RTS/CTS mechanism.

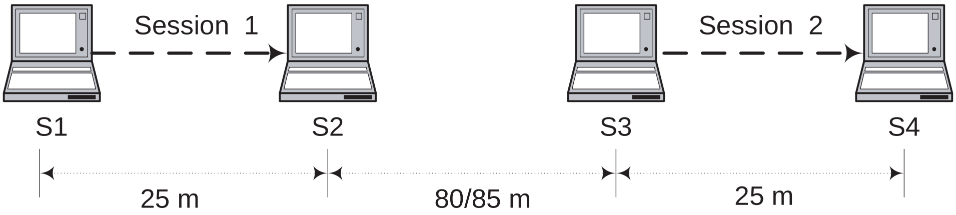


Figure 3.22. Network configuration at 11 Mbps.

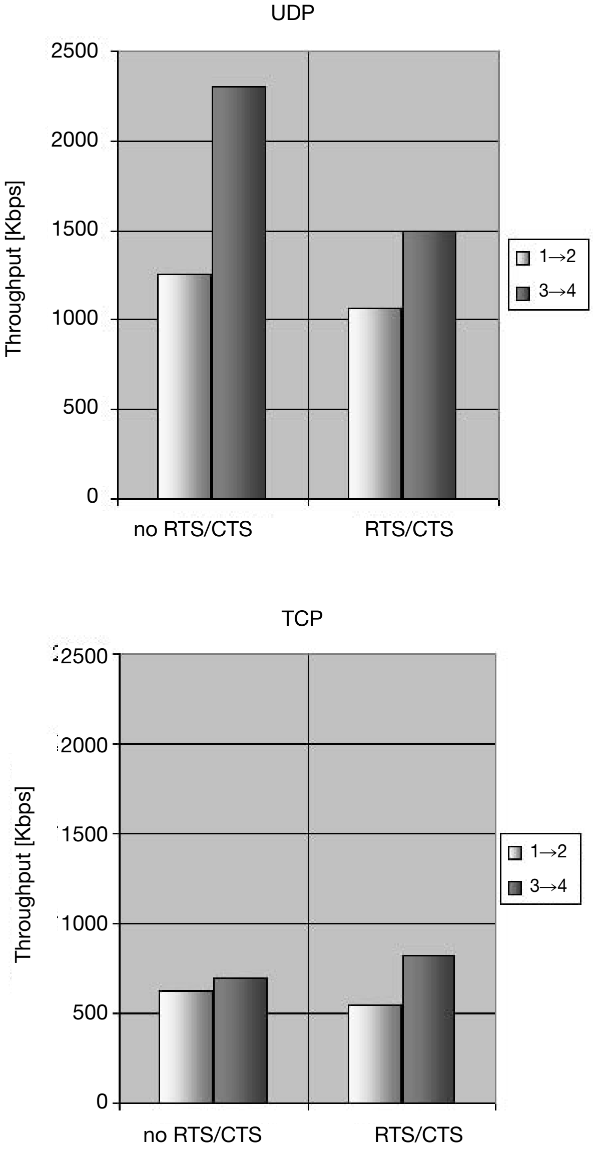


Figure 3.23. Throughputs at 11 Mbps.

mechanism, the S2 CTS forced S3 to defer the transmission of RTS frames (i.e., simply a delay in the transmission), whereas RTS frames sent by S3 forced S2 to not reply with a CTS frame to S1's RTS. In the latter case, S1 increased the backoff and rescheduled the transmission. Finally, when the TCP protocol was used, the differences between the throughput achieved by the two connections still existed but were reduced. The analysis of this case is very complex because we must also take into consideration the impact of the TCP mechanisms that (i) reduce the transmission rate of the first connection, and (ii) introduce the transmission of TCP-ACK frames (from S2 and S4), thus contributing to making the system less asymmetric.

3.5.4 Physical Carrier Sensing Range

Results presented in the previous section seem to indicate that dependencies among the stations extend far beyond the transmission range. For example, taking as a reference the scenario presented in Figure 3.22, the distance between the two couples of transmitting stations is about three times the transmission range. The hypothesis is that dependencies are due to a large physical carrier sensing that includes all the stations. To validate this hypothesis and to better understand the system behavior, we designed some experiments to estimate the physical carrier sensing range. A direct measure of this quantity seems difficult to achieve because the 802.11b cards we utilized do not provide to the higher layers information about the channel carrier sensing. Therefore, we defined an indirect way to perform these measurements. We utilized the scenario shown in Figure 3.24 with fixed distance between each coupled communicating stations [$d(1, 2) = d(3, 4) = 10$ meters], and variable distance between the two couples, that is, $d(2, 3)$ is variable.

The idea is to investigate the correlation among the two sessions while increasing the distance $d(2, 3)$. To measure the correlation degree, just before running each experiment we performed some preliminary measurements. Specifically, we measured the throughput of each session in isolation, that is, when the other session is not active. Then, we measured the throughput of each session when both sessions are active. Hereafter, $Th_i(x)$ denotes the throughput of session i ($i = 1, 2$) when both sessions are active and $d(2, 3) = x$. Obviously, $Th_i(\infty)$ denotes the throughput of session i($i = 1, 2$) when $d(2, 3) = \infty$, and, hence, the two sessions are independent. By exploiting these measurements, we estimated the correlation existing between the two sessions by the following index:

$$D_1(x) = 1 - \frac{Th_1(x) + Th_2(x)}{Th_1(\infty) + Th_2(\infty)}$$

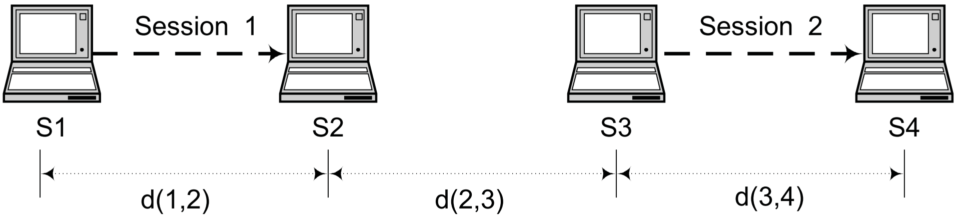


Figure 3.24. Reference network scenario.

The $D_1(x)$ index takes the value 0 if the two sessions are independent. Taken a session as a reference, the presence of the other session may have two possible effects on the performance of the reference session: (1) if the two sessions are within the same physical carrier sensing range, they share the same physical channel; (2) if they are outside the physical carrier sensing range, the radiated energy from one session may still affect the quality of the channel observed by the other session. As the radiated energy may travel over unlimited distances, we can expect that $D_1(x)$ may be equal to zero only for very large distances among the sessions [8].

When the $D_1(x)$ value is greater than zero, the index does not indicate how strong the correlation is. To measure this second aspect we introduce the $D_2(x)$ index:

$$D_2(x) = \frac{Th_1(0) + Th_2(0)}{Th_1(x) + Th_2(x)}$$

$D_2(x)$ compares the throughput of the two sessions when they are active at the same time and $d(2, 3) = x$, with respect to the two-session throughput when all the stations are inside the same transmission range, that is, $d(2, 3) = 0$. A $D_2(x)$ value equal to 1 indicates the maximum correlation that exists when all stations are in the same transmission range.

By varying the distance $d(2, 3)$ we performed several experiments to estimate the above indexes. The results were obtained with the card's transmission rates set to 2 and 11 Mbps, and are summarized in Table 3.7 and Table 3.8, respectively. As is clearly shown in the tables, the correlation among sessions is still marked when $d(2, 3)$ is less than or equal to 250 meters, noticeably decreases around 300 meters, and further decreases (but does not disappear) when the intersession distance is about 350 meters.

From the above results, we assume that 250 m is approximately the size of the physical carrier sensing range. After this distance, the correlation among the two sessions is due to the mutual impact on the channel quality. A set of measurements is currently being made to further verify the exact size of the physical carrier sensing range.

It is worth noting that the physical carrier sensing range is almost the same for the two different transmission rates. Indeed, the physical carrier sensing mainly depends on two parameters: the stations' transmitting power and the distance between transmitting stations. The rate at which data are transmitted has no significant effect on these parameters.

The results obtained confirm the hypotheses we made in the previous section to justify the apparent dependencies existing among the two couples of transmitting stations

Table 3.7. Throughput Values (Card Rate = 11 Mbps, Payload Size = 512 bytes)

		Throughput of Session 1		Throughput of Session 2			
Access Mechanism	Distance	$Th_1(\infty)$ Kbps	$Th_1(x)$ Kbps	$Th_2(\infty)$ Kbps	$Th_2(x)$ Kbps	$D_1(x)$	$D_2(x)$
	$x = 0$	2780	1849	2981	1768	0.37	1.00
	$x = 150$	1950	1500	2950	2250	0.23	0.96
No	$x = 180$	2920	2210	3040	1580	0.36	0.95
RTS/CTS	$x = 200$	2290	1930	3160	2660	0.16	0.78
	$x = 250$	2820	1700	3170	2760	0.25	0.81
	$x = 300$	2980	2800	3060	2750	0.08	0.65
	$x = 350$	2730	2590	3250	3230	0.03	0.62

Table 3.8. Throughput Values (Card Rate =2 Mbps, Payload Size = 512 bytes)

Access		Throughput Session 1		Throughput Session 2			
Mechanism	Distance	$Th_1(\infty)$	$Th_1(x)$	$Th_2(\infty)$	$Th_2(x)$	$D_1(x)$	$D_2(x)$
No RTS/CTS	$x = 0$	1279	577	1253	561	0.55	1.00
	$x = 150$	1310	880	1310	780	0.37	0.69
	$x = 180$	1310	930	1310	820	0.33	0.65
	$x = 200$	1270	1030	1330	1130	0.17	0.53
	$x = 250$	1300	960	1330	960	0.27	0.59
	$x = 300$	1370	1360	1380	1050	0.12	0.47
	$x = 350$	1360	1110	1400	1390	0.09	0.45

even if the distance between them is about three times greater than the transmission range.

It is worth noting that the ideal value for $D_1(0)$ is 0.5, that is, each session gets half of the throughput of the session in isolation. This is not true for CSMA MAC protocol as $Th_1(0)$ [$Th_2(0)$] is greater than $Th_1(\infty)/2$ [$Th_2(\infty)/2$]. This result is caused by the smaller overhead of the backoff algorithm in the experiments with $d(2, 3) = 0$.

3.5.5 Channel Model for an IEEE 802.11 Ad Hoc Network

The results presented in this paper indicate that for correctly understanding the behavior of an 802.11 network operating in ad hoc mode, several different ranges must be considered.

Specifically, as shown in Figure 3.25, given a transmitting station S, the stations around it will be affected by the station S transmissions in a different way depending on the distance from S and the rate used by S for its transmissions. Assuming that S is transmitting with a rate x ($x \in \{1, 2, 5.5, 11\}$), stations around it can be partitioned into three classes depending on their distance, d, from S:

1. Stations at a distance $d < TX\_Range(x)$ are able to correctly receive data from S, if S is transmitting at a rate lower or equal to x.
2. Stations at a distance d, where $TX\_Range(x) < d < PCS\_Range$, are not able to receive data correctly from station S. However, as they are in the S physical carrier sensing range, when S is transmitting they observe the channel busy and, thus, they defer their transmissions.
3. Stations at a distance $d > PCS\_Range$ do not measure any significant energy on the channel when S is transmitting, therefore they can start transmitting contemporarily to S; however, the quality of the channel they observe may be affected by the energy radiated by S. In addition, if $d < PCS\_Range + TX\_Range(x)$, some interference phenomena may occur (see below). This interference depends on the IF_Range value. This value is difficult to model and evaluate as it depends on several factors (mainly the power at the receiving site) but as explained before, $TX\_Range(1) < IF\_Range < PCS\_Range$.

Several interesting observations can be derived by taking into consideration points 1 to 3 above. Firstly, the hidden-station phenomenon, as it is usually defined in the literature (see

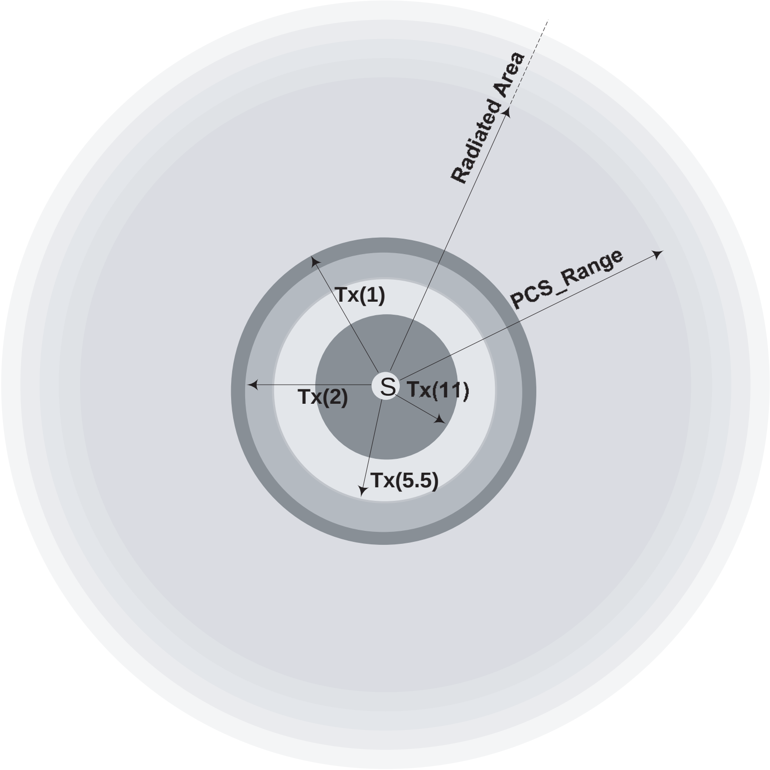


Figure 3.25. Channel model for an 802.11 ad hoc network.

Section 3.2.2), is almost impossible with the ranges measured in our experiments. Indeed, the *PCS_Range* is more than twice *TX_Range*(1) (the larger transmission range). Furthermore, two stations, say S1 and S2, that can start transmitting toward the same receiver, R, must be at a distance $\leq 2 \cdot TX\_Range(1)$, and thus they are inside the physical carrier sensing range of each other. Hence, if S1 has an ongoing transmission with R, S2 will observe a busy channel and, thus, will defer its own transmission. This means that in this scenario, virtual carrier sensing is not necessary and the RTS/CTS mechanism only introduces additional overhead.

Although the hidden-station phenomenon, as defined in the literature, does not seem relevant for this environment, point 3 above highlights that packets cannot be correctly received because of interference caused by a station that is "hidden" to the sending station. An example of this type of hidden-station phenomenon is presented in Figure 3.26. In this figure, we have two transmitting stations, S and S1, that are outside their respective *PCS_Range* and, hence, are hidden from each other. In addition, we assume that the receiver of station S (denoted by R in the figure) is inside the interference range (*IF_Range*) of station S1. In this scenario, S and S1 can be simultaneously transmitting and, if this occurs, station R cannot receive data from S correctly. Also in this case, the RTS/CTS mechanism does not provide any help and new coordination mechanisms need to be designed to extend the coordination in the channel access beyond the *PCS_Range*.

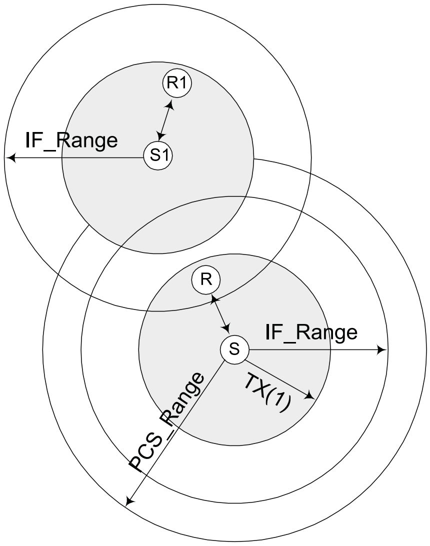


Figure 3.26. Interference-based hidden-station phenomenon.

It is worth noting that, in our channel model, the exposed-station definition (see Figure 3.6) must be modified, too. In this scenario, exposed stations are those station at a distance *PCS_Range*- *TX_Range*(1) < *d* < *PCS_Range*. Indeed, these stations are exposed to station S transmissions while they are in the transmission range of stations with *d* > *PCS_Range*. The following example outlines problems that may occur in this case. Let us denote by S1 a station at a distance *d* from S: *PCS_Range* < *d* < *PCS_Range* + *TX_Range*(*x*). Station S1 can start transmitting, with a rate *x*, toward a station E that is inside the physical carrier sensing area of S; station E cannot reply because it observes a busy channel due to the ongoing station S transmissions, that is, E is exposed to station S. Since station S1 does not receive any reply (802.11 ACK) from E, it assumes an error condition (collision or CRC error condition); hence, it back offs and then tries again. If this situation repeats several times (up to 7), S1 assumes that E is no longer in its transmission range, gives up the transmission attempt, and (wrongly) signals to the higher layer a link breakage condition, thus forcing higher layers to attempt a recovery action (e.g., new route discovery, etc.—see Section 3.3).

To summarize, results obtained in the configuration we analyzed indicate that the hidden-station and exposed-station definitions must be extended. These new hidden-station and exposed-station phenomena may produce undesirable effects that may degrade the performance of an ad hoc network, mainly if the TCP protocol is used. Extending the coordination in the channel access beyond the *PCS_Range* seems to be the correct direction for solving the above problems.

3.5.6 The Communication Gray Zones Problem

An important problem related to the different transmission ranges of control and data frames is the so-called *communication gray zones* problem [18]. This problem was revealed by a group of researchers at Uppsala University. While measuring the performance of their own implementation of the AODV routing protocol [22] in an IEEE 802.11b ad hoc network, they observed an unexpected large amount of packet losses, especially during route changes. They found that the increase in packet loss occurred in some specific geographic areas that they called "communication gray zones." In such zones, the packet loss experienced by a station may be extremely high—up to 100%—thus severely affecting the performance of those applications characterized by a continuous packet flow (e.g., file transfers and multimedia streaming). They also found that the ultimate reason for this phenomenon is that a station inside a gray zone is considered as reachable by a neighboring station, based on its routing information, but data communication between the stations is not possible. The same problem was found to affect other routing protocols like OLSR [5] and LUNAR [25].

To better understand why communication gray zones arise it is worthwhile to briefly recall how the AODV routing protocol works. AODV is a reactive protocol that discovers and maintains routes on demand. When a route to a target station is needed, the AODV protocol broadcasts a route-request message that is then disseminated throughout the network. When the target station (or a valid route to the target station) is found, a route-reply message is sent back to the requesting station by means of a unicast message. While this message travels towards the requesting station, routes are set up inside routing tables of the traversed stations.

In addition to the request–reply mechanism, the AODV protocol uses a sensing mechanism to discover neighboring stations and, based on this, to update, add, or remove routes in the routing table. Periodically, each station broadcasts HELLO beacons. Upon reception of a HELLO message from a neighbor, a station becomes aware that the neighboring station is reachable and can, thus, be used to relay data transmissions. Routing tables are, thus, updated accordingly.

Several elements contribute to the occurrence of communication gray zones. In particular, the different properties of HELLO messages with respect to data messages play an important role. These properties, and their effects, are summarized below.

1. **Transmission rate.** Since HELLO beacons are broadcast messages, they are transmitted at the basic rate (2 Mbps). On the other hand, data packets (which are unicast) may be transmitted at 11 or 5.5 Mbps. Therefore, HELLO messages have a transmission range larger than data messages.
2. **No Acknowledgment.** In 802.11b, broadcast messages are transmitted without acknowledgement. Therefore, a station that receives a HELLO message from a neighboring has no indication whether transmission is possible even in the opposite direction, that is, there is no indication that the link is bidirectional.
3. **Packet size.** In general, HELLO messages are much smaller in size than data packets. As is well known, small packets have a lower probability of being affected by transmission errors, and minor chances of colliding with other packets. Therefore, it is more likely for a HELLO message to reach a receiver than a data packet, especially when the link quality is poor.

In addition to the above elements, the effects of fluctuating links need to be taken into account as well. At the border of the transmission range, the communication quality tends to be fluctuating. Under such conditions, it may happen that a station sporadically receives a HELLO message from a neighbor, but this does not imply that consistent communication between the stations is actually possible. Since the AODV protocol updates routing tables based on the neighboring sensing mechanism (i.e., based on the reception of a HELLO message) it may occur that stable and longer routes are replaced by shorter but unreliable ones.

Figure 3.27 depicts a scenario pointed out by the researchers at Uppsala University, in which communication gray zones can be experienced by the mobile station MS (see [18]). In this scenario, stations labeled as GW, FS1, and FS2 are static, while station MS moves forward and back as indicated in the figure. There is an active communication between the gateway Station GW and the mobile station MS. Depending on the physical position of the mobile station, the traffic from MS to GW (and vice versa) is routed trough one, two, or three hops via intermediate stations FS1 and FS2. Theoretically, the MS always has a route toward GW. However, while moving from the initial position to the rightmost position, MS will pass through two gray zones. Similarly, two gray zones will be traversed in the reverse path. In [18] it is shown that traversal of the gray zones is associated with time intervals during which MS experiences a packet loss of up to 100%. The duration of this time interval, as well as the peak value in the packet loss experienced by the mobile station, depends on the specific routing protocol.

Before proceeding, it is important to highlight that the communication gray zone problem cannot be revealed by using the current simulation tools (e.g., ns-2). Indeed, in the IEEE 802.11 model implemented by simulation tools, both unicast and broadcast transmissions are performed at 2 Mbps and, hence, they have the same transmission range. Furthermore, connectivity is modeled as on/off, that is, the communication becomes impossible as soon as the distance exceeds the transmission range.

In [18] the authors also propose some possible solutions for alleviating the communication gray zone problem, namely: (i) the exchange of the neighboring set (i.e., stations include their neighboring set in HELLO messages); (ii) the transmission of N-consecutive HELLO messages; and (iii) the introduction of a SNR threshold to discard weak control messages. They have also assessed, by means of experimental analysis, that the SNR-threshold approach is the most effective and it eliminates the effects of communication gray zones almost completely.

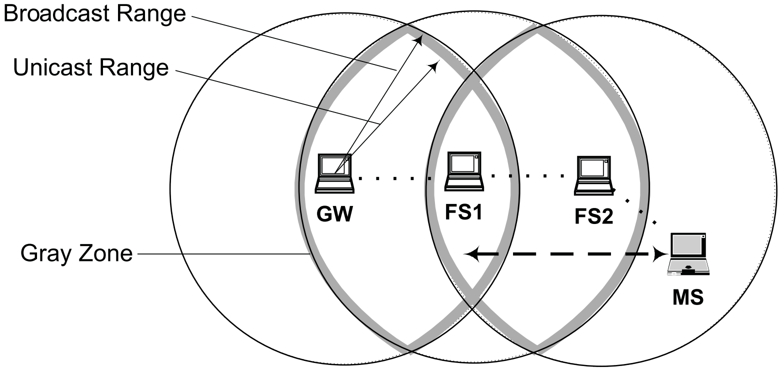


Figure 3.27. A scenario in which communication gray zones can be experienced (taken from [18]).

3.6 EVOLUTION OF IEEE 802.11b FOR AD HOC NETWORKS

In ad hoc networks, each station logically operates similarly to a router. However, from the physical standpoint, there is a significant difference between a router and a station in an ad hoc network. Typically, a router has multiple network interfaces, and a packet received from one interface is retransmitted through a different interface (see the left side of Figure 3.28). On the other hand, in a multihop ad hoc network a station has a single wireless interface and packets are received from and transmitted through the same interface (see the right side of Figure 3.28).

Current ad hoc network architectures do not take into account this difference, and implement in ad hoc stations the same functionalities of a router. Specifically, packets received from the wireless medium are delivered to the IP layer where a route lookup is performed based on the destination IP address (steps 1–3 in the left-hand part of Figure 3.29). If the packet is not destined to the station itself, it is passed down to the network interface to be retransmitted (steps 4 and 5 in the left-hand part of Figure 3.29).

The difference, from the forwarding standpoint, between an ad hoc station and a router has been recently pointed out by Acharya et al. [1], who have also proposed an architecture for efficient packet forwarding at stations in multihop ad hoc networks.

The first question addressed in [1] is which is the best architecture for forwarding a packet in ad hoc networks. The answer is highlighted in Figure 3.29. The left-hand side of the figure depicts the legacy approach for forwarding packets; the right-hand side shows the new approach proposed by the authors. In the latter case, the forwarding is completely managed by the network interface card (NIC). Upon receiving a packet, the NIC (by exploiting some local information) determines whether or not the packet has to be retransmitted. Only packets destined to the station itself are passed to higher-level protocols. Due to its behavior, the proposed architecture has been named cut-through architecture [1].

The cut-through architecture provides several advantages that can be classified into two categories. The first category includes advantages that are not related to a specific MAC protocol, where advantages belonging to the second category are strongly related to the random access scheme and the RTS-CTS mechanism used in the IEEE 802.11 MAC protocol. The following advantages belong to the first category:

1. The NIC does not need to interrupt the CPU for packet processing. This could lead to a considerable power saving if, for example, the station is used only for packet forwarding purposes. The CPU needs to wake up only for processing route updates.
2. Delays for transferring data from the NIC to the host, and vice versa, are avoided.

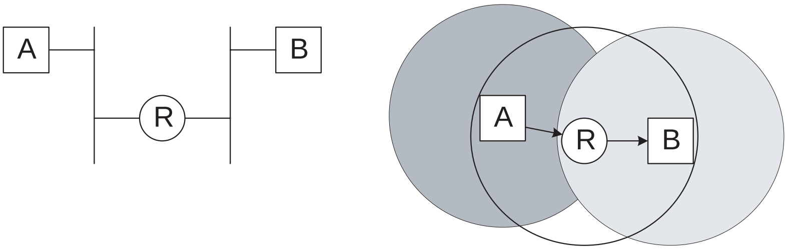


Figure 3.28. Packet forwarding in a router in a wired network (left) and in an ad hoc network (right).

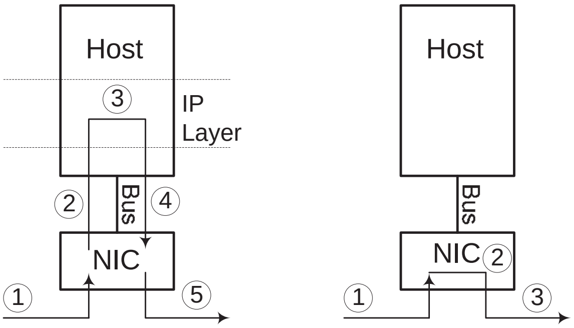


Figure 3.29. Forwarding in ad hoc stations: legacy approach (left) and NIC-based forwarding (right).

3. Local traffic does not further delay forwarding traffic. In legacy IEEE802.11 architecture, at the forwarding station, the packet is transferred to the main memory by the NIC. The host CPU is notified (e.g., via interrupts) for further processing of the packet by the IP protocol stack running on the host CPU. The host software (IP protocol stack) would typically queue up the packet in a transmission queue (together with the locally generated packets) and select packets for transmission based on a scheduling algorithm (typically, FIFO). Thus, packets generated by applications running at the station can overtake packets to be forwarded. This produces an increase in the end-to-end delay.

As the other advantages are strictly related to the IEEE 802.11 MAC protocol, they can be better understood by first considering the operations performed at the MAC layer by stations A, B, and R in the scenario depicted in the right side of Figure 3.28.

3.6.1 Forwarding Operation: Legacy approach versus NIC-Based Approach

Let us consider the case shown in Figure 3.30, where A is the upstream station, R is the forwarding station, and B the downstream station. In the legacy approach, the data delivery from A to B involves two separate and independent transmissions. For each transmission, the IEEE 802.11 MAC protocol (including the RTS/CTS mechanism) is used.

Specifically, with reference to Figure 3.30, the transmission from A to R is first performed. At R, the packet is passed to the IP protocol, processed by the IP software, and passed down to the NIC for transmission to B. At this time, R has to repeat the same procedure executed by A during the transmission to R. Note that the two transmissions (from A to R and from R to B) are independent of each other from the channel access standpoint.

It is worth noting that after the first RTS/CTS exchange, stations A and R has control of the channel and no other station in the transmission range of A and/or R can access the channel. However, this control is immediately lost after the ACK transmission from R to A. Clearly, from the Station R standpoint, it is not very wise to release the channel control and immediately after compete again for gaining control of the channel. It would be better for R to maintain in exclusive control of the channel. In this case, the transmission to B would be done without contention, thus improving the bandwidth utilization (there would

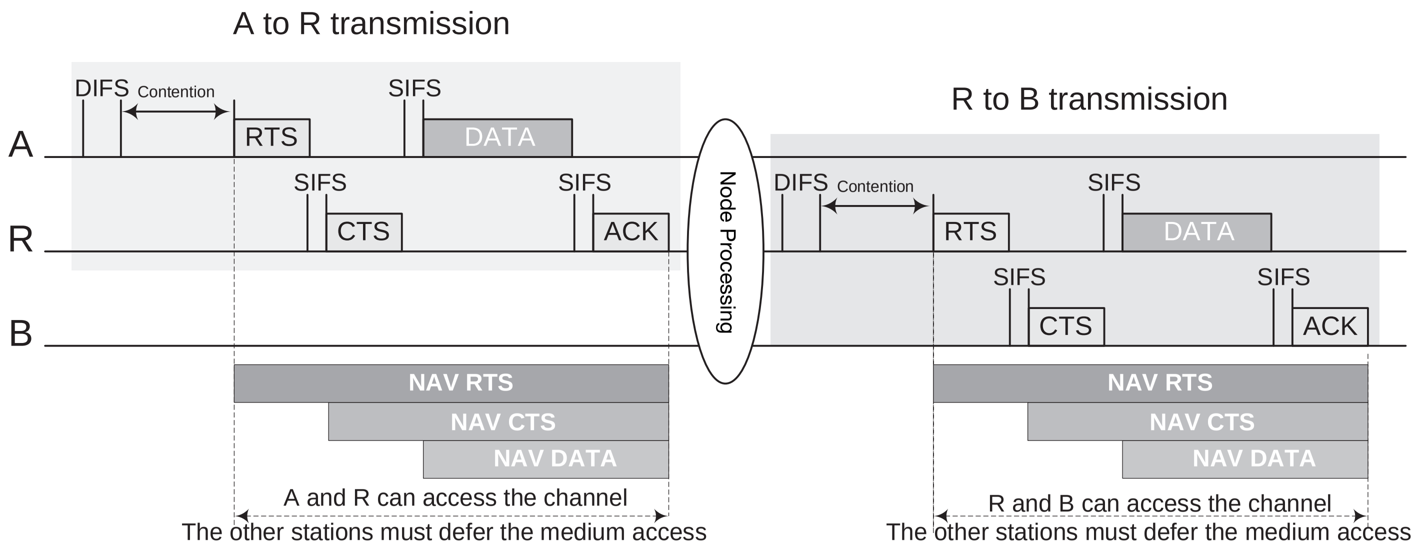


Figure 3.30. Forwarding operations in the legacy approach.

be no bandwidth wastage due to collisions and backoff periods) and minimizing the forwarding delay. Obviously, Station R should execute the packet forwarding very quickly so that the transmission from R to B could start immediately after the ACK transmission from R to A. This can be achieved only if the forwarding operation is performed completely inside the NIC (right-hand side of Figure 3.29).

Figure 3.31 shows an extension of the IEEE 802.11 MAC protocol to manage packet forwarding in a more efficient way [1]. The basic idea is to give the highest priority to the traffic to be forwarded by extending the channel reservation scheme to allow on-the-fly transmissions. Specifically, upon receiving a frame from A, Station R not only sends back an ACK frame to A but, at the same time, transmits an RTS frame to further extend the channel reservation.[12] Since the RTS frame transmission occurs while all the other stations within R's transmission range are still blocked (due to the previous RTS/CTS exchange), station R can immediately get the channel. The extended MAC protocol has been named Data-driven Cut-through Multiple Access (DCMA).

3.6.2 DCMA MAC Protocol

The DCMA MAC protocol is an extension of the IEEE 802.11 DCF and, as such, it follows the associated four-way handshake involving the exchange of RTS/CTS/DATA/ACK frames. As shown in the previous section, the DCMA attempts to replace the two distinct channel accesses (upstream and downstream) with a combined access. Specifically, DCMA combines the ACK (to the upstream station) with the RTS (to the downstream station) in a single ACK/RTS packet that is sent to the MAC broadcast destination address.

The cut-through approach, proposed in DCMA, fails when the downstream station (e.g., B in our example) cannot reply to the ACK/RTS (with a positive CTS). In such a case, the forwarding station then simply queues the packet in the NIC queue, and resumes the normal IEEE 802.11 channel access method using the exponential backoff to regulate subsequent access to the shared channel. The channel-contention resolution of DCMA is same as that of 802.11, with a station remaining silent as long as any of its one-hop neighbors is either receiving or transmitting a data packet. Accordingly, this protocol does not suffer from any additional penalties, over and above those present in 802.11.

Since DCMA has no notion of future reservations (all access attempts are for immediate transfer of DATA frames), it does not require any modifications or enhancements to the 802.11 NAV; a station simply stays quiet as long as it is aware of (contiguous) activity involving one or more of its neighbours. Any station that overhears an ACK/RTS not addressed to it merely increments the NAV by the time interval included in the ACK/RTS message.

In [1] a simulation analysis of the DCMA scheme is also presented. This analysis was carried out by implementing the DCMA access protocol in the ns-2 simulation tool. Consequently, the ns-2 typical values were used: bit rate of 2 Mbps, transmission range equal to 250 m, and interfering range equal to 550 m. All transmissions, regardless of the frame size, were preceded by an RTS/CTS exchange.

In the simulation study, high-rate sources were used to guarantee a never-empty queue at the transmitter. To avoid the interference of TCP mechanisms, the UDP protocol was used at the transport layer. The statistics were estimated by considering only the packets correctly received at the receiver. The forwarding station's routing tables were preconfig-

[12] More precisely, the RTS frame is piggybacked to the ACK frame.

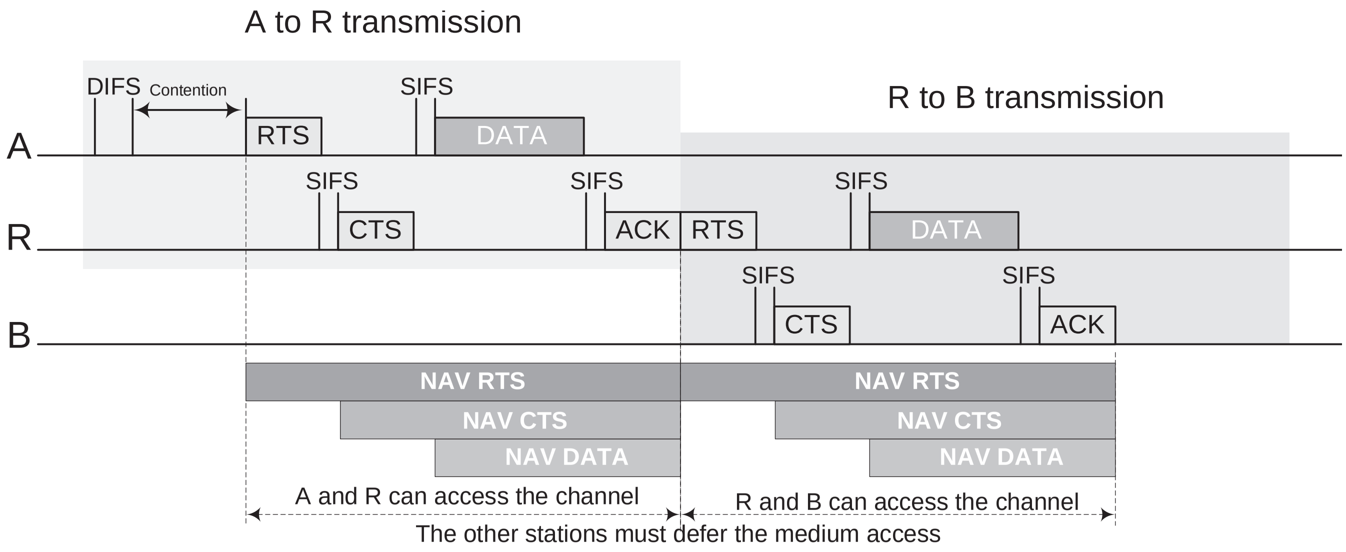

A to R transmission
R to B transmission
DIFS
Contention
SIFS
RTS
DATA
CTS
ACK
A
R
B
NAV RTS
NAV CTS
NAV DATA
A and R can access the channel
R and B can access the channel
The other stations must defer the medium access

Figure 3.31. Forwarding operations in the NIC-based approach.

Table 3.9. Throughput Comparison (in Kbps)

	Packet size (bytes)			
	256	512	1024	1536
802.11	159	200	231	258
DCMA	197	242	282	301

Table 3.10. End-to-End Delay Comparison (in sec)

	Packet size (bytes)			
	256	512	1024	1536
802.11	1.00	1.67	2.59	2.83
DCMA	0.50	0.81	1.31	1.73

ured with the shortest-path routes to their respective destinations. The contents of these routing tables will be briefly discussed in the following paragraphs. A complete description can be found in [1].

Several configurations are considered in [1]. For the sake of brevity, only one set of them are discussed here. They refer to a string or chain topology (see Figure 3.8) in which the distance between successive stations is 250 m. A single flow of UDP packets is transmitted from the leftmost to the rightmost station. Several experiments were conducted by varying the size of the payload from 256 to 1536 bytes.

The results obtained by considering a 7-hop chain are summarized in Table 3.9 and Table 3.10. It clearly appears that DCMA improves the throughput by around 20% with respect to the standard protocol, whereas the delay improvement is more significant, ranging from 63% (1536 bytes) to 100% (256 byte packets).

In [1], the comparison was further extended by considering an increasing number of hops in the chain. The results obtained are consistent with results presented in Table 3.9 and Table 3.10: the delay reductions with DCMA are significant (on the order of 50%), whereas throughput improvements are marginal.

Finally, the influence of the offered load was considered. The results obtained are summarized in Table 3.11 and Table 3.12. Specifically, these results are related to a 12-hop chain and have been obtained by increasing the sending rate at the source from 250 Kbps to about 500 Kbps. It clearly appears that there are different saturation points for the two protocols. The IEEE 802.11 MAC protocol has the maximum throughput at around 0.375 Mbps; after this offered load level, the queues start to build up, the end-to-end delay

Table 3.11. Throughput (in Kbps) as a Function of the Offered Load

	Offered Load (Kbps)										
	250	275	300	325	350	375	400	425	450	475	500
802.11	242	267	292	316	340	364	259	273	256	241	235
DCMA	242	267	292	316	340	364	389	413	376	376	374

Table 3.12. End-to-End Delay (in sec) as a Function of the Offered Load

	Offered Load (Kbps)										
	250	275	300	325	350	375	400	425	450	475	500
802.11	0.075	0.075	0.075	0.075	0.075	0.075	0.075	0.075	0.75	0.95	1.09
DCMA	0.080	0.080	0.080	0.080	0.080	0.080	1.37	1.53	2.06	2.02	2.08

shows a significant increase, and the throughput decreases. On the other hand, DCMA has the maximum throughput at around 0.425 Mbps. Furthermore, after the saturation point DCMA shows a more stable behavior: the throughputs remains high, and the end-to-end delay is about half that of the standard protocol.

ACKNOWLEDGMENTS

This work was funded partially by the Information Society Technologies Programme of the European Commission, Future and Emerging Technologies, under the IST-2001-38113 MobileMAN project, and partially by the Italian Ministry for Education and Scientific Research in the framework of the FIRB-VICOM project. The authors wish to express their gratitude to Veronica Vanni, Eleonora Borgia, and Riccardo Bettarini for performing the experiments discussed in this chapter.

REFERENCES

1. A. Acharya, A. Misra, and S. Bensal, "A Label-switching Packet Forwarding Architecture for Multi-hop Wireless LANs," in *Proceedings of the ACM Workshop on Mobile Multimedia (WoWMoM 2002),* Atlanta, GA, September 28, 2002.
2. A. Ahuja et al., "Performance of TCP over Different Routing Protocols in Mobile Ad-Hoc Networks," in *Proceedings of the IEEE Vehicular Technology Conference (VTC 2000),* Tokyo, Japan, May 2000.
3. G. Anastasi , E. Borgia, M. Conti, and E. Gregori, "IEEE 802.11 Ad Hoc Networks: Performance Measurements," in *Proceedings of the Workshop on Mobile and Wireless Networks (MWN 2003),* Providence, Rhode Island, May 19, 2003.
4. K. Chandran, S. Raghunathan, S. Venkatesan, and R. Prakash, "A Feedback Based Scheme for Improving TCP Performance in Ad Hoc Wireless Networks," *IEEE Personal Communication Magazine,* Special Issue on Ad Hoc Networks, *8,* 1, 34–39, February 2001.
5. T. Clausen et al., "Optimized Link State Routing Protocol," Internet Draft, IETF MANET Working Group, November 2002, available at http://menetou.inria.fr/olsr/draft-ietf-manet-olsr-07.txt.
6. T. D. Dyer and R. V. Boppana "A Comparison of TCP Performance over Three Routing Protocols for Mobile Ad Hoc Networks," in *Proceedings of the ACM Symposium on Mobile Ad Hoc Networking & Computing (MobiHoc),* October 2001.
7. T. Ephremides, "A Wireless Link Perspective in Mobile Networking," ACM Mobicom 2002 keynote speech, available at http://www.acm.org/sigmobile/mobicom/2002/program/.
8. T. Ephremides, "Energy Concerns in Wireless Networks," *IEEE Wireless Communications,* 48–59, August 2002.

9. Z. Fu, X. Meng, and S. Lu, "How Bad TCP Can Perform in Mobile Ad Hoc Networks," in *Proceedings of the IEEE Symposium on Computers and Communications (ISCC 2002),* Taormina-Giardini Naxos, Italy, July 2002, pp. 298–303.
10. Z. Fu, P. Zerfos, K. Xu, H. Luo, S. Lu, L. Zhang, and M. Gerla, "The Impact of Multihop Wireless Channel on TCP Throughput and Loss,"*Proceedings of IEEE INFOCOM 2003,* San Francisco (CA), March 30 - April 3, 2003.
11. GloMoSim, Global Mobile Information Systems Simulation Library, http://pcl.cs.ucla.edu/projects/glomosim/.
12. G. Holland and N. Vaidya, "Analysis of the TCP Performance over Mobile Ad Hoc Networks," in *Proceedings of the ACM International Conference on Mobile Computing and Networking (MobiCom'99),* Seattle, WA, August 1999, pp. 207–218.
13. G. Holland and Nitin H. Vaidya "Analysis of TCP Performance over Mobile Ad Hoc Networks," *ACM/Kluwer Journal of Wireless Networks 8*(2–3), 275–288, 2002.
14. Official Homepage of the IEEE 802.11 Working Group, http://grouper.ieee.org/groups/802/11/.
15. IEEE Standard 802.11, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," August 1999.
16. J. Li, C. Blake, D. De Couto, H. Lee, and R. Morris, "Capacity of Wireless Ad Hoc Wireless Networks," in *Proceedings of the ACM International Conference on Mobile Computing and Networking (MobiCom'01),* Rome, Italy, pp. 61–69, July 2001.
17. J. Liu and S. Singh, "ATCP: TCP for Mobile Ad Hoc Networks," *IEEE Journal on Selected Areas in Communications (J-SAC), 19*(7), 1300–1315, July 2001.
18. H. Lundgren, E. Nordstron, and C. Tschudin, "Coping with Communication Gray Zones in IEEE 802.11 Based Ad Hoc Networks," in *Proceedings of the ACM Workshop on Mobile Multimedia (WoWMoM 2002),* Atlanta, GA, September 28, 2002, pp. 49–55.
19. J. P. Monks, P. Sinha, and V. Bharghavan, "Limitations of TCP-ELFN for Ad Hoc Networks," in *Proceedings of MoMuc 2000,* Tokyo, Japan, October 2000.
20. The Network Simulator–ns-2. http://www.isi.edu/nsnam/ns/index.html.
21. M. S. Obaidat and D. G. Green, "An Accurate Line of Sight Propagation Performance Model for Ad Hoc 802.11 Wireless LAN (WLAN) Devices," in *Proceedings ICC 2002,* New York City, April 28–May 2, 2002.
22. C. Perkins and E. Royer, "Ad Hoc On-Demand Distance Vector Routing," in *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications (WMCSA'99),* February 1999.
23. Qualnet simulator, http://www.qualnet.com/.
24. K. Tang and M. Gerla, "Fair Sharing of MAC under TCP in Wireless Ad Hoc Networks," in *Proceedings of IEEE MMT'99,* Venice (I), October 1999.
25. C. Tschudin and R. Gold, "LUNAR: Lightway Underlay Network Ad Hoc Routing," available at http://www.docs.uu.se/docs/research/projects/selnet/lunar/lunar.pdf.
26. S. Xu and T. Saadawi, "Does the IEEE 802.11 MAC protocol Work Well in Multihop Wireless Ad Hoc Networks?" *IEEE Communication Magazine, 39,* 6, 130–137, June 2001.
27. S. Xu and T. Saadawi, "Revealing the Problems with 802.11 MAC Protocol in Multi-hop Wireless Networks," *Computer Networks, 38,* 4, March 2002.
28. K. Xu, S. Bae, S. Lee, and M. Gerla, "TCP Behavior across Multihop Wireless Networks and the Wired Networks," in *Proceedings of the ACM Workshop on Mobile Multimedia (WoWMoM 2002),* Atlanta (GA), September 28, 2002, pp. 41–48.
29. K. Xu and M. Gerla, "TCP over an IEEE 802.11 Ad Hoc Network: Unfairness Problems and Solutions," UCLA Computer Science Deptartment Technical Report—020019, May 2002.
30. G. Zaruba and S. Das, "Off-the-Shelf Enablers of Ad Hoc Networks," in *Mobile Ad Hoc Net-*

working, S. Basagni, M. Conti, S. Giordano, and I. Stojmenovic (Eds.), IEEE Press/Wiley, Hoboken, NJ, 2004.

31. Feng Wang and Yongguang Zhang, "Improving TCP Performance over Mobile Ad-Hoc Networks with Out-of-Order Detection and Response," in *Proceedings of ACM MobiHoc 2002,* Lausanne, Switzerland, 2002.

CHAPTER 4

SCATTERNET FORMATION IN BLUETOOTH NETWORKS

STEFANO BASAGNI, RAFFAELE BRUNO, and CHIARA PETRIOLI

4.1 INTRODUCTION

It is widely anticipated that fourth-generation wireless systems will extensively rely on the unlicensed operations provided by *ad hoc communications* [1]. Allowing spontaneous deployment and self-planning/management, ad hoc networking will play an important role in delivering all kinds of wireless services from the Internet to the very hands of the mobile user.

The Bluetooth (BT) technology, as described in the Specifications of the Bluetooth System, Version 1.1 [2], is expected to be one of the most promising enabling technologies for ad hoc networks. Originally introduced as short-range cable replacement, the BT specifications define ways that each BT device can set up multiple connections with neighboring devices so that communication can be established in a multihop fashion. In this sense, Bluetooth devices spread out in a geographic area can provide the missing wireless extension to the various heterogeneous network infrastructures, allowing a more pervasive wireless access.

This chapter describes solutions to the fundamental problems that need to be addressed for the self-organization of Bluetooth devices into an ad hoc network.

According to the specifications, when two BT nodes that are in each others' communication range want to set up a communication link, one of them must assume the role of *master* of the communication while the other becomes its *slave*. This simple "one-hop" network is called a *piconet* and may include several slaves, no more than seven of which can be actively communicating with the master at the same time. If a master has more than seven slaves, some slaves have to be "parked." To communicate with a parked slave, a master has to "unpark" it, while possibly parking another slave.

Mobile Ad Hoc Networking. Edited by Basagni, Conti, Giordano, and Stojmenovic.
ISBN 0-471-37313-3 © 2004 Institute of Electrical and Electronics Engineers, Inc.

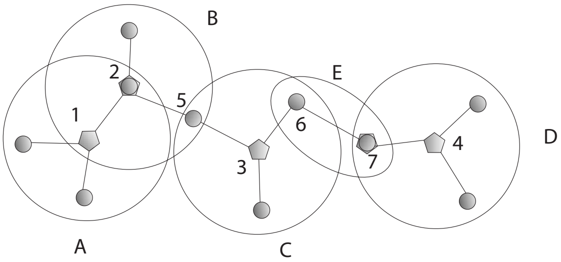


Figure 4.1. Five piconets forming a connected scatternet.

The specifications allow each node to assume multiple roles. A node can be a master in one piconet and a slave in one or more other piconets, or a slave in multiple piconets. Devices with multiple roles act as *gateways* to adjacent piconets, thus creating a multihop ad hoc network called a *scatternet*.

Figure 4.1 shows the case in which 13 BT devices have been partitioned into four piconets (A, B, C, and D). Masters are represented by pentagons (surrounded by a large circle that represents their transmission radius), whereas slaves are depicted as small circles. Adjacent piconets can be interconnected in different ways. Piconets A and B depict the *master–master* case, when two masters are neighbors and interconnection is achieved by having one of the two masters joining the piconet of the other as a slave (in the figure, node 2 became the slave of node 1). Two piconets can be joined by a common slave, termed a *gateway slave*. This is the case of piconets B and C, which are joined by node 5. The third case is when piconets are interconnected through a pair of neighboring slaves, called in the following *intermediate gateways,* as in the case of piconets C and D, joined by nodes 6 and 7. In the latter case, interconnection requires that one of the two intermediate gateways becomes the master of a new piconet that includes the other intermediate gateway as a slave (in the figure, node 7 becomes the master of the extra piconet). With the creation of piconet E, the five piconets of Figure 4.1 form a connected scatternet.

In this chapter, we describe the solutions proposed so far for scatternet formation. The next section introduces to the basics of the Bluetooth technology. In Section 4.3, we define the problems underlying scatternet formation, and we also sum up the desirable properties that should be satisfied by a generated scatternet. Section 4.4 surveys solutions that have appeared so far in the literature. Sections 4.5 and 4.6 describe in detail two protocols for generating connected scatternets. Section 4.7 describes some concerns raised by the implementation of some of the protocols according to the current version of the Bluetooth specifications. Finally, Section 4.8 concludes the chapter.

4.2 BLUETOOTH BASICS

In this section, we briefly describe the procedures of the Bluetooth technology that are needed to describe solutions for scatternet formation. This section is not intended to provide a detailed description of the Bluetooth system, for which the reader is referred to [2].

Bluetooth operates in the 2.4 GHz, unlicensed ISM band. Frequency-hopping spread spectrum technology is adopted to reduce interference both among BT nodes and with technologies that operate in the same band such as IEEE 802.11b.

In order to establish a connection between two BT nodes, one of them assumes the role of *master* of the communication and the other one becomes its *slave*. This simple "one hop" network is called a *piconet* and may include many slaves, no more than seven of which can be active at the same time. All devices in a piconet share the same channel (i.e., a frequency-hopping sequence) which is derived from the unique ID and Bluetooth clock of the master.

Communication to and from a device is always performed through the master of the piconet to which it belongs. In particular, a Time-Division Duplex (TDD) scheme is employed for intrapiconet communications: transmissions occur in pairs of 625 μs slots, the first of which is for master–slave communication, and the second for the communication from the polled slave to the master.

A BT device can time share among different piconets. In particular, a device can be either the master of one piconet and a slave in other piconets or a slave in multiple piconets. A node with multiple roles acts as *gateway* between the piconets to which it belongs. Piconets can be interconnected through gateways into a multihop ad hoc network called a *scatternet*.

Piconet formation is performed in two steps: first, devices must become aware of their neighboring nodes, that is, the nodes in their transmission range (device discovery); then, information must be exchanged to set up a link between a candidate slave and a candidate master (link establishment). According to the current BT specifications, the former step is accomplished by means of the *inquiry* and *inquiry scan* procedures, whereas the latter requires the *page* and *page scan* procedures.

For device discovery to happen, two neighboring devices have to be in "opposite" modes, namely one must be the inquirer (the discovering device), and the other device has to be willing to be discovered. These modes are implemented in BT by having the inquirer be in inquiry mode, and the other device be in inquiry scan mode. The inquirer transmits inquiry ID packets asking neighboring devices to identify themselves and to provide synchronization information needed for link establishment at a later time. To minimize the device discovery time, the BT specifications state that ID packets must be very small (i.e., they include only the General Inquiry Access Code, GIAC, and nothing else) and that they must be transmitted over the frequencies of a predefined inquiry/inquiry scan frequency hopping sequence, changing frequencies at a high rate (twice a slot). A device in inquiry scan hops among different frequencies at a very low rate (one frequency every 1.28 s), thus increasing the probability of a handshake on the same frequency of the inquirer. As soon as an ID packet is received at a device in the inquiry scan mode, the device computes a backoff interval and starts listening again. Only when an ID packet is received after the backoff phase, the unit in inquiry scan mode will send an FHS (Frequency Hop Synchronization) packet containing its identity and synchronization information (its BT clock).

The described inquiry procedures lead to an asymmetric knowledge of two neighboring devices: The inquirer identity is not known at the device that received an inquiry ID packet. After successful reply from the device in the inquiry scan mode, the inquirer instead, knows the identity and the clock of the neighbor that just replied. This enables the inquirer v to estimate the frequency hopping sequence used by its neighbor and thus to invite it to join its piconet as a slave. This invitation is accomplished by means of the paging procedures.

In order for two neighboring devices u and v to establish a link, one must be in page mode (for instance, node v) and the other in page scan mode (node u). By definition, the device that is in page mode is the master. Node v transmits a page ID packet on u's frequencies, containing *u*'s address. When *u*, which is in page scan mode, receives such a packet, it immediately acknowledges it. At this point, *v* transmits to *u* a FHS packet that bears all the required information for *u* to synchronize on *v*'s own frequency-hopping sequence. Finally, the two devices exchange all the information for setting up a link and a piconet is formed with *v* as the master and *u* as its slave.

It may happen that device *u*, which is in page scan, is already the master of another piconet and that it could host *v* as one of its slaves. In this case, once a piconet has been established between *v* and *u*, with *v* as the master, the slave *u* can request a *switch* of role. This situation is explicitly addressed by the BT specifications, and it is implemented via exchanging a specific Link Manager Protocol (LMP) packet that instructs the two devices to switch to the frequency-hopping sequence of the new master.

To save the energy of BT devices, "low-power operation modes" have been included in the specifications that allow BT nodes to "go to sleep" when they are not actively involved in communication. This feature is also used to let a master "release" a slave so that the slave can perform protocol-related operations in another piconet. Among the several modes provided in the specifications for low-power operations, we outline here the functioning of the *park mode*. A slave that has been put in park mode by its master cannot be actively involved in communication with that master. However, parked slaves periodically wake up in predefined beacon slots to listen to their master communication. Unparking of (possibly multiple) devices is achieved by transmitting an LMP unpark Protocol Data Unit (PDU) in the beacon slot. This packet carries the ID of the devices to be unparked and their new active slave addresses. Parked slaves can trigger an unpark LMP PDU by sending explicit requests during preallocated slots (access window). Similarly, active devices can ask to be parked (or they can be parked by their master) by exchanging an LMP park packet with their master.

4.3 PROBLEM DESCRIPTION

The problem of scatternet formation concerns the grouping of the network nodes into piconets (*piconet formation*), and the joining of the piconets into a connected scatternet (*piconet interconnection*). These operations require each node to be aware of its neighbors. A phase of *device discovery* must, therefore, be performed before the actual scatternet formation process takes place. For these operations, we describe here their desirable features and the barriers to their implementation.

In what follows, given a set of BT nodes, we call a *visibility graph* the network topology in which there is a link between any two nodes whose Euclidean distance is less than or equal to the nodes' transmission radius (for the sake of protocol description, we assume that all nodes have the same transmission radius). These topologies are also often referred to as *unit disk graphs* [3].

4.3.1 Device Discovery

The device discovery phase should lead each of the network nodes to become aware of all its neighbors in the visibility graph. This neighbor knowledge should be "symmetric," which means that if node *v* knows node *u*, *u* must also know *v*. In general, unless simplifying assumptions are made on the visibility graph (e.g., the visibility graph is a clique, as in "single-hop" topologies), this is the least information required for performing the following phases of piconet formation and interconnection. As described in Section 4.2, the mechanisms provided by the BT specifications for device discovery (inquiry procedures) do not lead to the needed symmetric neighbor knowledge and require nodes to be in opposite inquiry modes in order to be able to communicate. Therefore, specifications-compliant mechanisms must be defined to ensure that, for each pair of neighboring nodes v and u, they are eventually in opposite modes and that, when node v discovers node u, u is also made aware of v.

The implementation of the device-discovery mechanisms as outlined above is challenged by a number of BT standard features. We mentioned already the asymmetry introduced by the inquiry procedures: since the inquirer does not transmit its unique BT address, a node that receives an inquiry ID packet cannot discern if this packet comes from an already discovered node. This leads to useless inquiry handshakes, and may compromise the possibility of knowing the entire neighborhood. In addition, ensuring that two nodes are in opposite modes (for instance, by having them alternating between inquiry and inquiry scan mode) can be guaranteed only statistically, and it is a time-consuming process. This is also exacerbated by the duration of the backoff interval, which makes the inquiry handshake long.

4.3.2 Piconet Formation

The piconet formation phase concerns the assignment of roles—either master or slave—to all the network nodes. As dictated by the specifications, if a node is a master, it is master only in one piconet. In general, a slave can be enrolled in more than one piconet.

Desirable properties for piconet formation include:

- Distributed operations. Networks of Bluetooth devices are characterized by high dynamics (e.g., because of mobility and nodes joining the network at different times), and gathering complete information about the changing network topology can be infeasible. Therefore, piconet formation should be executed at each node with limited knowledge of the node's surrounding topology. One-hop or two-hop neighborhood knowledge is what can be known at each node in reasonable time and with limited resource consumption.
- Piconet size limited to eight nodes. Since no more than seven slaves can be actively communicating with a master, a master of a piconet with more than seven slaves is forced to park and unpark its slaves in order for all of them to be able to communicate. Parking and unparking have an associated overhead both in terms of induced delay and bandwidth. The throughput available to the nodes is significantly reduced in the case of large piconets, leading to inefficient operations.
- Resource-based master selection. Being a master is more resource consuming than being a slave, as masters have to handle and coordinate all the communications to and from all the nodes in the piconet. Piconet formation should, therefore, be performed, taking into account the different types of devices and their available resources when assigning the role of master.

4.3.3 Piconet Interconnection

The final phase concerns the selection of *gateway devices* to interconnect multiple piconets into a scatternet. There are three ways of interconnecting two piconets:

1. *Master–master*. In the case in which the masters of the two piconets are neighbors, interconnection can be achieved by having one of the two masters join the other piconet as a slave.
2. *Gateway slaves*. When a slave is neighbor of two masters, whether that slave belongs to only one of them or to both, it may be used to interconnect the two piconets by joining the piconet to which it does not belong (if any). When this is the case, that slave is called a gateway slave.
3. *Intermediate gateways*. This is the case in which two masters have two slaves that are neighbors. The interconnection of the two piconets can be achieved by requiring that one of the two gateways becomes the master of a new piconet that includes the other intermediate gateway as slave.

Desirable characteristic of piconet interconnection (which are properties of the resulting scatternet) are:

- Connected scatternet. If the topology resulting from the device discovery phase is connected, the scatternet generated by a scatternet formation protocol should also be connected.
- Resilience to disconnections in the network. If the topology resulting from the device discovery phase is not connected, a scatternet formation protocol should be able to correctly operate in the connected components of the network.
- Routing robustness. The scatternet should have multiple routes between any pairs of nodes.
- Limited route length. Scatternet routes are longer than the corresponding routes in the topology resulting from the device discovery phase for two main reasons:
    1. All intrapiconet communications pass through a master (two neighboring slaves that belong to the same piconet cannot communicate directly).
    2. The piconet-based network organization, which, for instance, may force nodes that are one hop away but that do not belong to the same piconet to communicate through a much longer interpiconet route.

Scatternet formation protocols should carefully select gateways so that the increase in the route length is limited.

- Selection of gateway slaves. Whenever possible, gateway slaves are to be preferred for piconet interconnection. This is due to the fact that when a master is also a slave for interconnection purposes, when it acts as a slave, all the communications to and from all the nodes in its own piconet are “frozen,” which clearly detrimentally affects throughput performance.
- Small number of roles per node. A node should have the minimum number of roles. If a node has x roles, then it belongs to x different piconets. Switching between two piconets has an overhead due to synchronization to the frequency hopping sequence of the current master.
- Self-healing. When the network topology varies dynamically (due to nodes mobility, arrival of new nodes, failures of links/nodes, etc.), a scatternet formation protocol should be able to converge to a scatternet that retains all the properties of the initial scatternet.

Papers that have addressed desirable properties of scatternet formation protocols along with the identification of key metrics for their performance evaluation are [4], [5], [6], and [7].

4.4 OVERVIEW OF PROPOSED SOLUTIONS

The BT specifications describe methods for device discovery and for the participation of a node in multiple piconets. However, solutions for scatternet formation are not provided.

A first broader classification of the solutions proposed so far in the literature distinguishes between scatternet formation protocols that require the radio vicinity of *all* nodes (*single-hop* topologies) and protocols that work in the more general *multihop* scenario. The solutions are usually distributed and localized, in the sense that the protocols are executed at each node with limited knowledge of the surrounding topology.

4.4.1 Single-Hop Solutions: Device Discovery and Scatternet Formation

In some of the scatternet formation protocols for single-hop topologies, the device-discovery phase takes place concurrently with the phases of piconet formation and interconnection. Therefore, in this section the three phases are described together. In all cases, the nodes involved in the device discovery process keep switching between the inquiry and inquiry scan modes. This mechanism allows two nodes to eventually be in opposite modes at the same time, which is the needed condition for them to meet each other.

Among the solutions that work only in single-hop topologies with n nodes, the first algorithm for scatternet formation has been presented in [8]. The *Bluetooth Topology Construction Protocol* (BTCP) described in that paper is based on a distributed leader-election process. The leader election is performed by means of the inquiry and inquiry scan procedures so that when two neighboring nodes discover each other, one of the two nodes "wins," that is, it keeps participating in the discovery of other nodes, while the loser quits this phase, letting the winner know about its identity, its clock, and the identities and the clocks of all the nodes it was made aware of via previous confrontations. The elected leader (final winner) will eventually know the number, identities, and clocks of *all* the nodes in the network, based on which it decides the role that each node performs in the final scatternet. The computed roles are then communicated by the leader to all nodes. Designated masters are informed of the list of their designated slaves. The specific centralized algorithm, performed locally by the leader aims at minimizing the number of piconets while generating a mesh-like connected scatternet whose piconets have no more than seven slaves and are interconnected by gateways of degree two. The centralized algorithm executed by the leader generate a connected scatternet that satisfies the required properties only when the number of nodes in the network is ≤ 36. When $n > 36$, other centralized schemes could be used such as the one proposed in [9], which also takes traffic into account. The problem of scatternet formation is formulated as an integer linear programming problem where the objective function to be minimized is the traffic load at the most congested node (bottleneck node). Another centralized scheme has been presented in [10], which works for networks with any number of nodes. The aim of the protocol is to design a scatternet topology for which optimal interpiconet scheduling can be defined and that obtains max–min fairness. Once it has gathered all the information about all the network nodes, the leader computes a k-regular topology (i.e., a topology in which all nodes have k links). The final topology is obtained by selecting and combining k different disjoint sets of edges that are (near-) perfect matchings (also called 1-factors), $2 \leq k \leq 2n - 1$. The authors describe a selection of the 1-factors that leads to a connected topology.

A randomized distributed algorithm for single-hop topologies is described in [11]. The protocol proceeds in rounds (i.e., it is a synchronous protocol). The devices are grouped into components (that may be either a single device, a piconet, or a connected scatternet), each of which has a leader. They are progressively joined to form the final connected scatternet. In every round, leaders of different components attempt to discover each other. This is performed by having each leader randomly enter either the inquiry or inquiry scan mode for that round. Leaders in opposite modes that discover each other decide which of the nodes in their components they can be interconnected with, thus forming a larger component. The protocol terminates when all the nodes belong to the same component. The resulting scatternet has the following properties: Each piconet has no more than seven slaves, and every gateway has degree two. The number of generated piconets is close to the theoretical minimum $\lceil n/7 \rceil$. The scatternet, which is a tree, is formed in $O(\log n)$ rounds with high probability.

A tree-like scatternet is also produced in the Tree Scatternet Formation (TSF) protocol described in [12]. The idea behind the protocol is very similar to the one presented in [11], although TSF is an asynchronous protocol. At any point in time, the scatternet being generated by TSF is a forest of connected trees. As the protocol proceeds, trees are joined by their roots that, by periodically alternating between inquiry and inquiry scan mode, try to discover each other. Since the trees' interconnection happens only via the roots, and since any node could be a root, all nodes must be in the transmission range of each other for the generated scatternet to be connected (single-hop solution). Newly arriving nodes can be included in already formed trees, which render this solution self-healing, that is, able to cope with changes in the network topology. The solution aims at minimizing the number of piconets, and at keeping the number of nodes per piconet below seven. However, this cannot be always guaranteed. In general, generating a tree-like scatternet topology simplifies routing but introduces limits in terms of robustness and efficiency.

4.4.4 Multihop Solutions

Protocols for general multihop topologies rely on the assumption that each node is aware of its neighbors and this knowledge is symmetric. This knowledge is provided by the device discovery phase, which is, therefore, performed before the actual piconet formation and interconnection phases.

4.4.2.1 Device discovery. The solution for device discovery in multihop networks uses a mechanism introduced in [13] and [14] that is similar to that described in [8]. Each device alternates between inquiry mode and inquiry scan mode, remaining in each mode for a time selected randomly and uniformly in a predefined time range. The operations while in each of the two modes are those described in the specifications. When two nodes in opposite inquiry modes handshake, they set up a temporary piconet that lasts only the time necessary to exchange their ID and possibly other information necessary for the following phases of the protocol. The formation of temporary piconets and the exchange of information achieves the required mutual knowledge.

The following procedure describes the operations performed at each device v as it enters the topology discovery phase of the protocol:

```
Discovery($v$)
$T_{\text{disc}} \leftarrow \ell_{\text{td}}$
while $T_{\text{disc}} > 0$
   do if Rand(0. 1) < 0.5
      then InquiryMode
            Compute($T_{\text{inq}}$)
            Inquiry(min($T_{\text{inq}}$, $T_{\text{disc}}$))
            InquiryScanMode
            Compute($T_{\text{scan}}$)
            InquiryScan(min($T_{\text{scan}}$, $T_{\text{disc}}$))
      else InquiryScanMode
            Compute ($T_{\text{scan}}$)
            InquiryScan(min($T_{\text{scan}}$, $T_{\text{disc}}$))
            InquiryMode
            Compute($T_{\text{scan}}$)
            Inquiry(min($T_{\text{inq}}$, $T_{\text{disc}}$))
exit
```

The generic device v that executes the discovery procedure sets a timer T_{disc} to a predefined time length of the discovery phase ℓ_{td}. This timer is decremented at each clock tick (T_{disc} keeps track of the remaining time until the end of this phase).

Device v then randomly enters either the inquiry or inquiry scan mode, and computes the length of the selected phase (T_{inq} or T_{scan}). This computation is performed by randomly and uniformly selecting the phase duration in a predefined interval. While in a given mode, device v performs the inquiry procedures as described by the BT specifications. The procedures that implement the inquiry mode (procedure inquiry) or the inquiry scan mode (procedure inquiryScan) are executed for the computed time (T_{inq} and T_{scan}, respectively), not to exceed T_{disc}. Upon completion of an inquiry (inquiry scan) phase, a device switches to the inquiry scan (inquiry) mode. A node keeps alternating between the two opposite modes until $T_{\text{disc}} > 0$. As mentioned, to allow each pair of neighboring devices to achieve a mutual knowledge of each other, our scheme requires that whenever a device in inquiry (inquiry scan) mode receives (sends) an FHS packet, a temporary piconet is set up by means of a page phase, and devices exchange their ID and possibly other information. As soon as this information has been successfully communicated, the piconet is disrupted.

The effectiveness of the described mechanism in providing the needed mutual knowledge to pairs of neighboring devices relies on the idea that by alternating between inquiry and inquiry scan mode, and randomly selecting the length of each inquiry (inquiry scan) phase, we have high probability that any pair of neighboring devices will be in opposite mode for a sufficiently long time, thus allowing the devices to discover each other.

The duration of the discovery phase should be chosen so that each node is made aware of enough of its neighbors to guarantee network connectivity. This implies that, as opposed to single-hop solutions, in multihop topologies when two nodes discover each other they both have to keep performing device discovery as there might be other nodes that need to discover them to be connected to the rest of the network (and vice versa).

4.4.2.2 Scatternet Formation. Among the solutions that apply to the more general case of multihop topologies, the scatternet formation protocol described in [15] requires that the protocol be initiated by a designated node (the *blueroot*) and generate a tree-like scatternet. The blueroot starts the formation procedure by acquiring as slaves its one-hop neighbors. These, in turn, start paging their own neighbors (those nodes that are at most two hops from the root) and so on, in a "wave expansion" fashion, until the whole tree is constructed. In order to limit the number of slaves, it is observed that if a node in a unit disk graph has more than five neighbors, then at least two of them must be connected. This observation is used to reconfigure the tree so that each master node has no more than seven slaves. If a master v has more than seven slaves, it selects two of them that are necessarily connected and instructs one of the two to be the master of the other, which is then disconnected from v's piconet. Such branch reorganization is carried throughout the network, leading to a scatternet in which each piconet has no more than seven slaves. Depending on a selected node to start the formation procedure, this solution does not work in the case of networks whose topology after the discovery phase is not connected. Furthermore, the implementation of the protocol requires the use of time-outs to solve possible deadlocks in the piconet formation phase.

Solutions for scatternet formation in multihop BT networks that produce topologies different from a tree are those presented in [14], [16], [17], [18], and [19].

The protocol presented in [13] and [14] proceeds from the device discovery phase as described above into the *BlueStars* (i.e., piconet) formation phase, and the configuration of the BlueStars into the connected scatternet. The phase of piconet formation deploys a clustering-based approach for master selection [20]. Based on a locally and dynamically computed weight (a number that expresses how suitable that node is for becoming a master), each node decides whether it is going to be a master or a slave. This phase starts at some dynamically selected nodes and terminates with the formation of disjoint piconets, each with one master and possibly multiple slaves. The final phase concerns the selection of *gateway devices* to connect multiple piconets so that the resulting scatternet is connected.

This solution has the following features. It works for general multihop BT networks. The generated scatternet is a mesh with multiple paths between any pair of nodes. The selection of the BT masters is driven by the suitability of a node to be the "best fit" for serving as a master. The generated scatternet is connected whenever the network resulting from the device discovery phase is connected. Finally, even in case of a disconnected discovered topology, the protocol generates a connected scatternet over each connected component.

The scatternet formation scheme proposed in [16], *BlueNet,* produces a scatternet whose piconets have a bounded number k of slaves. After the device discovery phase, each node randomly enters the page or the page scan mode with probability p (phase 0). When a node succeeds in getting at least one slave, it proceeds to phase 1 and tries to acquire up to k neighboring nodes as slaves. Otherwise, it keeps randomly entering the page or page scan mode and executing phase 0 until all neighboring nodes have communicated that they joined some other node's piconet. (The fact that a node in phase 0 can actually contact all its neighbors can be guaranteed only statistically.) In case a phase 0 node remains isolated, it enters phase 2, goes to page mode, and tries to interconnect to neighboring piconets by acquiring as slaves one node from each such piconet (up to k). After having accomplished this task, a phase 2 node exits the protocol. A master in phase 1 that has contacted all its neighbors and acquired at most k nodes in its piconet, proceeds to phase 3, the piconet interconnection phase. In this phase, the slaves of the piconets formed in phase 1, by alternating between page and page scan mode, attempt to set up links with neighboring slaves of other phase 1 piconets as instructed by their masters. The connectivity of the resulting scatternet is not guaranteed (i.e., not all the BlueNets are connected, even when the topologies resulting from the discovery phase are).

The main aim of the protocol proposed in [17] and [18] is to build up a connected scatternet in which each piconet has no more than seven slaves. To this purpose, degree reduction techniques are initially applied to the network topology graph to reduce the number of wireless links at each node to less than seven without disconnecting the network. Any (multihop) scatternet formation protocol can then be executed on the resulting topology, yielding to a scatternet whose piconet size is at most eight (one master and at most seven slaves). These techniques require each node to be equipped with additional hardware that provides the node with its current (geographic) location (e.g., a GPS receiver). Details of this solution, combined with the BlueStars protocol outlined above, are given in Section 4.5.

The idea behind *BlueMesh,* the scatternet formation protocol presented in [19] and [21], is to generate a connected scatternet by selecting some masters among the network nodes, and allowing each master to select *at most* seven slaves. The selection of the slaves is performed in such a way that if a master has more than seven neighbors, it chooses seven slaves among them so that via them it can reach all the others. Once masters and slaves are selected, i.e., piconets are formed throughout the network, gateways are chosen so that there is an interpiconet route between all masters that are at most tree hops away (i.e., all adjacent piconets are interconnected). This condition ensures the connectivity of the BlueMesh scatternet [22]. Further details about BlueMesh are given in Section 4.6. BlueMesh improves previous solutions in that: a) as opposed to BlueTrees, the generated scatternet is a more robust mesh and it also works in connected components of a possibly disconnected network; b) all generated scatternets are connected, which is not the case with BlueNet; c) as opposed to BlueStars, no piconet in a BlueMesh scatternet has more than seven slaves; and finally, d) no extra hardware is required as in the geometric-based solutions of [18].

Thorough performance evaluation of BlueStars has been presented in [23]. A performance comparison of the solutions for multihop scatternet formation presented in [14], [16], and [18] is given in [24].

4.5 GEOMETRIC TECHNIQUES AND SCATTERNET FORMATION

In this section, we describe the details of a scatternet formation protocol that produces scatternets that are connected and whose piconets have a bounded number k of slaves (usually it will be $k = 7$).

The protocol assumes that each node knows its own identity, a dynamically computed *weight* that indicates how much that node is suitable for serving as a master, and its own location in the plane (usually provided by an on-board GPS device, or by any suitable inertial positioning system device). It is also assumed that, as the outcome of the device discovery phase, a node also knows the identity of its neighbors, their weight, and their location.

For the sake of clarity, in the description of the algorithm we assume that nodes are randomly and uniformly scattered in the plane and that the network graph resulting from the device discovery phase is a connected unit disk graph (UDG).

The knowledge of the location is exploited for applying to the UDG geometric-based techniques to reduce the degree of the network to at most k. Once a connected topology with such a bounded degree has been obtained, the BlueStars algorithm for scatternet formation outlined above uses the nodes' weight for selecting the masters, the slaves, and the gateways necessary to form a degree-bounded connected scatternet.

In [18], several degree reduction techniques are described, and it is proven that the resulting degree-bounded topologies are connected. Here we describe only one of those techniques, namely the one that [18] deems the most promising for the Bluetooth technology, termed *Yao construction*. This technique was first proposed by Yao to construct the minimum spanning tree of the graph originated by a set of points in high dimension efficiently [25]. The reader is referred to [18] for a description of other geometric techniques for degree reduction in wireless networks modeled by UDGs.

The Yao construction is executed at each node *v* and proceeds as follows. Node *v* divides the plane that surrounds it into *k* equal angles. In each angle, node *v* chooses the closest neighbor *u*, if any. (Ties are broken arbitrarily.) A link between nodes *v* and *u* survives the Yao contruction phase if and only if *v* has chosen *u* and vice versa. All other links are deleted. To make such decision, nodes need to exchange with their neighbors the information on the nodes they selected. The mechanism we use for information exchange is the temporary setup of a piconet between every pair of neighboring nodes. For this to be possible, we have to guarantee that every pair of nodes are in opposite page modes. This is obtained by having the nodes execute the following protocol. Upon completing the local selection of links, a node *v* checks whether it has the biggest weight among its neighbors *N*(*v*). If this is the case, that node, called in the rest of the chapter an *init* node, executes the following procedure:

```
PeckOder($v$)
  PageMode
  for each smaller $u$ in $N(v)$
     do Page($u, v$)
  exit
```

An init node goes into page mode and starts paging all its neighbors (which, by definition, are "smaller neighbors," i.e., nodes with a smaller weight) setting up temporary piconets with each one of them. The information exchanged in the temporary piconet concerns whether the two nodes have chosen each other or not.

Symmetrically, a noninit node *u* executes the following procedure:

```
SubNode($u$)
  PageScanMode
  for each bigger $u$ in $N(u)$
     do waitPage($u, v$)
  PeckOrder($u$)
```

Node *u* goes into page scan mode and waits for a page from all its neighbors with bigger weights ("bigger neighbors"). As soon as node *u* has decided about all the links with the bigger neighbors (i.e., it has been paged by all of them), it goes to the "bottom of the pecking order"; that is, being now the biggest node among those with which it has to exchange the link information, it switches to page mode, and starts setting up temporary piconets with all its smaller neighbors (if any).

The topology resulting from the Yao construction, as described, is connected and has the property that no node has more than *k* neighbors [18].

Once a connected topology with such a bounded degree has been obtained, the BlueStars algorithm for scatternet formation outlined in Section 4.4 uses the nodes' weight for selecting the masters, the slaves and the gateways necessary to form a degree-bounded connected scatternet.

Let us consider the network of Fig. 4.2 as the network resulting from the device discovery phase. The only node with more than seven neighbors is node 36. Therefore, node 36 executes the Yao construction procedure to discard one of its eight neighbors. Assuming that nodes 20 and 21 fall in the same seven angles in which node 36 has partitioned the plane around itself, the result of the Yao construction phase is the cancellation of the link between node 36 and node 21. At this point, a connected scatternet is obtained by executing BlueStars over the "Yao topology" just obtained (where now nodes 36 and 21 are no longer neighbors). BlueStars, described in [13] and [14], proceeds from the Yao topology to the following two phases of piconet formation and interconnection of the piconets into a connected scatternet. Based on a locally and dynamically computed weight (a number that expresses how suitable that node is for becoming a master) and on the knowledge of the weight of its neighbors (obtained during the discovery phase and the Yao construction phase), each node decides whether it is going to be a master or a slave. This decision is taken at a node depending on the decision of the bigger neighbors, and is then communicated to the smaller neighbors. The mechanism through which this is implemented is similar to the pecking order protocol described above. In particular, a node that decided to be master is either an init node or a node whose bigger neighbors all decided to be slaves. A node that has been told (via paging) by one or more of its bigger neighbors that they are masters, becomes the slave of the first master that paged it. This phase of the protocol leads to the partition of the topology resulting from the discovery phase and the Yao construction into piconets. BlueStars does not guarantee a bounded number of slaves per piconet. However, its combination with the Yao contraction (which limits the nodal degree to *k*) also obtains this desirable property. The execution of the piconet formation phase of BlueStars over the Yao topology obtained from the topology depicted in Fig. 4.2 is shown in Fig. 4.3.

Nodes 21, 30, and 36, being init nodes, start paging their neighbors, which are all in page scan mode. As depicted in Figure 4.3, node 21 is successful in paging nodes 17 and 20, which become part of its piconet. Node 30 has no competitors in having nodes 11 and 22 join its piconet. Node 36 forms the largest piconet by acquiring nodes 6, 8, 13, 15, and 25 as slaves. Once nodes 13, 22, and 25 have communicated their decision to become slaves, node 7, whose bigger neighbors are all slaves, decides to be a master,

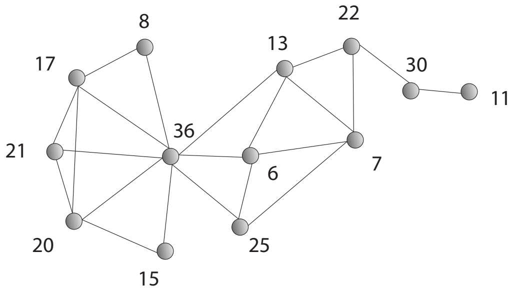


Figure 4.2. A Bluetooth network after the discovery phase.

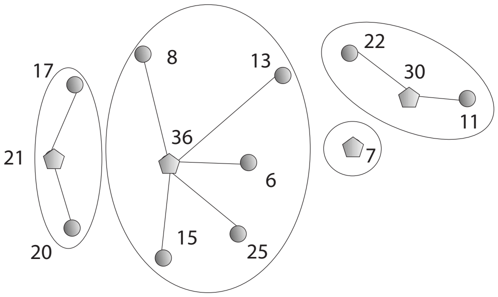


Figure 4.3. BlueStars' piconet formation.

and since node 6 is already affiliated to node 36, it will be a master of a piconet with one node.

After the piconet formation phase, each master proceeds to the selection of gateway devices to connect multiple piconets so that the resulting scatternet is connected. In order to achieve connectivity, it is necessary (and sufficient) that each master establishes a path with (i.e., chooses gateways to) all the masters that are at most three hops away [22]. The knowledge about which nodes are the masters two and three hops apart is achieved during the piconet formation phase. Specifically, each node *v* communicates its role (and possibly the identity and weight of its master) to all its smaller neighbors and to the bigger neighbors that became slaves. If a node is a slave, it waits for the smaller neighbors to communicate the same information. In this way, at the end of the piconet formation phase each node is aware of the identity of all its neighbors, and of the identity and weight of its masters, which is the information needed in the piconet interconnection phase. The process of piconet interconnection is based again on a mechanism similar to the pecking order protocol, this time executed only among the masters. The result of the BlueStars piconet interconnection phase over the network of piconets of Figure 4.3 is depicted in Figure 4.4.

By the end of the piconet formation phase, node 36 knows the identity and the weight of all the master at most three hops away, and the slaves through which it can reach them. For instance, one (or more) among nodes 6, 13, and 25 could be used as gateway slaves to interconnect to node 7. Once it has chosen one of these nodes (say, the biggest one: node

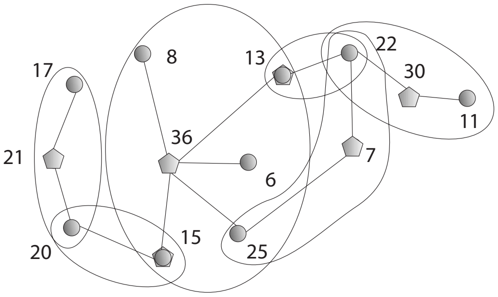


Figure 4.4. The scatternet produced by combining the Yao construction with BlueStars.

25), it instructs it to wait for a page from node 7. Node 7 also knows that nodes 6, 13, and 25 are the nodes through which the two piconets can be interconnected, and it adopts the same gateway selection rule of node 36. Therefore, it pages node 25 to enroll it as its slave. The same thing happens between masters 7 and 30, which select as intermediate slave node 22 (the sole node that can fulfill this purpose). The piconets of nodes 36 and 21 must interconnect via intermediate slaves. To this purpose, since both masters know each other and which of their slaves are neighbors of the slaves of the other, they can consistently choose two neighboring slaves and instruct them to page each other in order to form the needed new piconet. In the case depicted in Fig. 4.4, for instance, node 36 instructs node 15 (the biggest of its slaves that are neighbors of slaves of node 21) to page node 20, which is its neighbor that is a slave of node 21. Consistently, node 21 instructs its slave 20 to go into page scan mode and waits for a page from node 15. Similarly, the piconets of masters 36 and 30 are joined by the new piconet formed by node 13 and 22, which act as intermediate slaves. (The details of the rule adopted for consistent gateway selection and for piconet interconnection can be found in [14].)

4.6. BLUEMESH

In this section, we describe BlueMesh, a protocol for forming connected scatternets whose piconets have a bounded number of slaves. As with BlueStars, scatternet connectivity is guaranteed by establishing an interpiconet route between any two masters that are at most three hops away [22, Theorem 1]. Unlike the protocol presented in the previous section, BlueMesh does not need location information.

BlueMesh proceeds in successive iterations. Each iteration is executed by the network nodes that have not yet exited the execution of the protocol at some previous iteration. Let us call $G_i = (V_i, E_i)$ the network topology graph at iteration i, $i \geq 1$. G_1 is simply the topology after the device discovery phase (as before, we assume that this topology is a UDG). Each of the G_i, $i > 1$, is the subgraph of G_1 that spans the nodes of V_1 that did not exit the execution of BlueMesh in one of the previous iterations. In each iteration i, piconets are formed from the nodes in the topology graph G_i. The interconnection of two piconets is achieved either via a gateway slave or via a pair of intermediate gateways, one of which belongs to one piconet and the other to the other piconet. Gateway slaves are selected in the current iteration so that the piconets they belong to are joined. Masters then proceed to select the intermediate gateways between adjacent piconets not yet interconnected.

The intermediate gateways are the nodes that proceed onto the next iteration. All masters, slaves that have not been selected as gateways, and the gateway slaves exit the execution of BlueMesh at this time. BlueMesh terminates when all nodes have exited the execution of the protocol.

The functioning of BlueMesh is illustrated by the following example. We assume that BT devices know their identity, their weight, and the identities and weights of all their one-hop and two-hop neighbors. Two-hop neighbor knowledge can be achieved after the device discovery by executing the "pecking order" protocol described in Section 4.5, where the information exchanged via the temporary piconet between the nodes v and u are the lists of neighbors $N(v)$ and $N(u)$.

Each iteration of the protocol is performed locally at each node v and is made up of two parts: *role selection* (for piconet formation) and *gateway selection*.

Role selection is executed by every node at the very beginning of each iteration i (in the case of the first iteration role selection is performed as soon as the two-hop neighbor discovery process has been completed). Based on its weight and the weight of its one-hop neighbors, a node determines whether it is an init node in G_i. Only init nodes go into page mode. All the other nodes go into page scan mode.

Let us consider again the network of Fig. 4.2. Being the nodes with the biggest weight in their neighborhood, devices 30 and 36 are init nodes. They are masters and switch to page mode. All the other nodes go into page scan mode. Device 30, which has less than seven neighbors, selects all of them (nodes 11 and 22) as its slaves and pages them to communicate its decision. "Piconet 30" is then formed by nodes 11, 22 and 30. Device 36 has eight neighbors. Since we want to form piconets each with a number of slaves $\leq k = 7$, node 36 must select only seven of its eight slaves. Slave selection is performed at a master node v by executing the following procedure, where $S(v)$ is the set of selected neighbors and $C(v)$ denotes the set of v's bigger neighbors that are slaves and v's smaller neighbors.

```
ComputeS(v)
1  $S(v) \leftarrow \emptyset$
2  $U \leftarrow C(v)$
3  while $U \neq \emptyset$
4      do $x \leftarrow$ bigger in $U(v)$
5          $S(v) \leftarrow S(v) \cup \{x\}$
6          $U \leftarrow U \setminus N(x)$
7  $S(v) \leftarrow S(v) \cup$ Get$(7 - |S(v)|, C(v) \setminus S(v))$
```

Each master v chooses as slaves those neighbors in $C(v)$ that "cover" all the other neighbors in the sense that if a neighbor u is not selected as v's slave, then at least one of u's neighbors has been selected by v. Such coverage is always possible by selecting at most five slaves [19].

Procedure ComputeS implements a greedy approach for computing $S(v)$. One of v's neighbors in $C(v)$ (e.g., the one with the biggest weight) is selected as its slave, say node x. The procedure is then executed again on the set of all nodes in $C(v) \setminus \{x\}$, which are not covered by x. This rule allows node v to select up to five of its neighbors in $C(v)$ through which all other neighbors can be reached.

The function Get(m, W) returns a set of m nodes from the set W (for instance, the m smaller ones, or randomly chosen ones, or the bigger ones, etc.). It is used by ComputeS for selecting additional slaves to a maximum of seven ($|S(v)| \leq 7$).

By executing ComputeS(36), node 36 first selects five nodes—devices 8, 13, 15, 21, and 25—through which 36 can reach all its remaining neighbors. Then, by selecting nodes 17 and 20 it reaches the limit of seven slaves (procedure Get, line 7). At this point, node 36 pages *all* its neighbors. It will communicate to node 6 that it is not invited to join its piconet, and it will invite all the other selected nodes. As opposed to what happens in BlueStars, when a node is invited to join a piconet, it always accepts the invitation even if it already belongs to other piconets. Piconet 36 is made up of eight devices: master 36 and the seven slaves 8, 13, 15, 17, 20, 21, and 25. In piconet 30, slave 22 has been paged by all its bigger neighbors. It switches to page mode and starts paging the nodes in $C(22)$, namely, devices 7 and 13, communicating its role. Similarly, device 25, which has received the page from node 36, pages nodes 6 and 7, which are in $C(25)$. Device 13 received the pages from all its bigger neighbors, and it is now ready to communicate its role (slave of master 36) to its smaller neighbors 6 and 7. Device 7, which now knows that all its bigger neighbors (nodes 13, 22, and 25) are slaves, becomes master and pages all of its four neighbors, inviting them to join its piconet, which they do. Piconet 7 is thus formed by its master and the slaves 6, 13, 22, and 25. At this point, the network has been divided into three piconets that need to be interconnected. This marks the start of the gateway selection part of the current iteration. In this part, all slaves communicate to their master(s) information about the roles of their neighbors, their neighbors' list of masters, and whether some of their neighboring masters selected them as slaves. The information is obtained with an extra network-wide "wave" of messages exchanged in a pecking-like fashion during the role selection phase. Based on this information, each master decides which slaves to select as gateways to which piconet in order to obtain a connected scatternet. If a pair of masters have selected common slaves, they choose the bigger one among them as the gateway slave. This is the preferred way to interconnect adjacent piconets. Whenever no gateway slave can be selected to interconnect adjacent piconets, intermediate gateways are selected, again based on their weight (e.g., so that the sum of their weights is maximized, or the minimum weight is maximized, or any other unambiguous selection rule). Upon completion of these operations, the gateway slaves, together with the masters and the non-gateway slaves, exit the execution of the BlueMesh protocol. The intermediate gateways proceed to iteration $i + 1$ to form new piconets that interconnect them, hence providing connectivity between the piconets they affiliated with in iteration i. Going back to our example, devices 22 and 25, common slaves of piconet 7 and 36, and 7 and 30, respectively, are selected as gateway slaves to interconnect such piconets. Piconets 30 and 36 can be interconnected only through the pair of intermediate gateways 13 and 22. These two nodes are the only two nodes that move to the next iteration of the protocol. All the other nodes quit the execution of BlueMesh at this time. Some of these nodes are masters (of a single piconet), some are just slaves in one piconet (as is node 6), and some have multiple roles (e.g., gateway slaves 13, 22, and 25). Realizing that it is now an init node, node 22 goes into page mode and pages its smaller neighbor 13, which is waiting for its page. The two-node piconet 22 is thus formed, which implements the interconnection between masters 36 and 30. The scatternet resulting from the execution of BlueMesh on the network of Fig. 4.2 is displayed in Fig. 4.5. Masters are depicted as pentagons, the piconets generated in the first iteration are star shaped, and the piconet generated in the second iteration has an oval shape.

4.7 IMPLEMENTATION CONCERNS

The main concerns about the implementation of most of the described protocols for scatternet formation in multihop networks are due to the discovery phase as we have described it in Section 4.4.2.1. In particular, we observed the following two problems:

First, it is extremely time consuming, and therefore impractical, to discover all the neighbors of a given node. We have simulated the device discovery phase by using a BT extension to the ns2 network simulator, which implements all the details of the BT protocol stack. The device discovery was run for a predefined time T_{disc} over each visibility graph generated by distributing uniformly and randomly n = 30, 50, 70, 90, 110 nodes over a square area of side L = 30 meters. The transmission range of each node was that of Power Class 3 BT nodes (10 m). Nodes alternate between inquiry and inquiry scan mode,

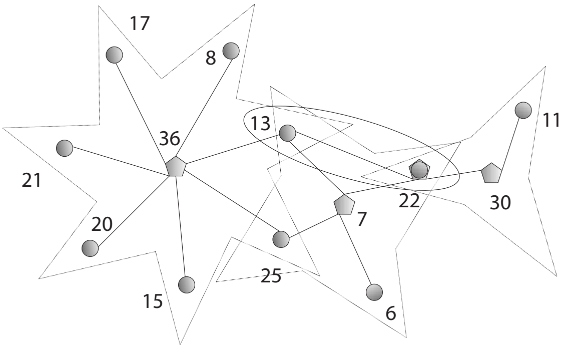


Figure 4.5. A BlueMesh scatternet.

spending a variable time, uniformly and randomly selected in the interval (0.02*s*, 2*s*), in each mode. The resulting topology, which we call a BT topology, has links only between those pairs of BT nodes that were able to discover each other during the device discovery phase.

We observe that it may take a long time to discover all the neighbors of a node (e.g., only 47% of a node's neighbors have been discovered after 10 s of device discovery in networks with 110 nodes). However, a shorter time suffices for discovering enough neighbors so that the resulting BT topologies are connected, which is the needed requirement for obtaining connected scatternets. This is shown by Figure 4.6. We can provide statistical guarantees that the BT topologies are connected in case of moderately dense to heavily dense visibility graphs provided that the device discovery runs for at least 6 s.

Three are the features of the Bluetooth technology that we have identified as having a strong impact on the device discovery duration: a) The need to adopt (stochastic) mechanisms to have neighboring nodes in opposite inquiry modes, so they can discover each other; b) the impossibility of identifying the inquirer, which demands the construction of a temporary piconet between neighbors that have discovered each other already; and c) the overly long duration of the backoff interval as stipulated in the BT specifications (2048 clock ticks). In our performance evaluation, we have quantified the impact of each of these features on the performance of device discovery [24]. We observe that, by just decreasing the backoff duration to one-fourth of the value specified by the standard, a significant increase in the number of discovered neighbors is achieved that leads to connected topologies in less than 2 seconds. For instance, in networks with 110 nodes—by far the worst case in the simulated scenarios—over 80% of a node's neighbors are discovered within 10 seconds.

Second, the topology resulting from the device discovery phase may not be a UDG graph. This violates the assumption on which all protocols that provide connected scatternets with piconets with less than $k = 7$ slaves rely. For instance, it might no longer be true that given a node v with more than five neighbors, it is possible to find two among v's neighbors that are physically neighbors and have discovered each other. For the protocol to work correctly, after the device discovery phase, it is, therefore, necessary to perform an extra phase leading to a consistent knowledge of each node's neighborhood. Basically, we want two nodes that have discovered a common neighbor and are neighbors themselves to discover each other. This can be obtained in the following way. At the end of the

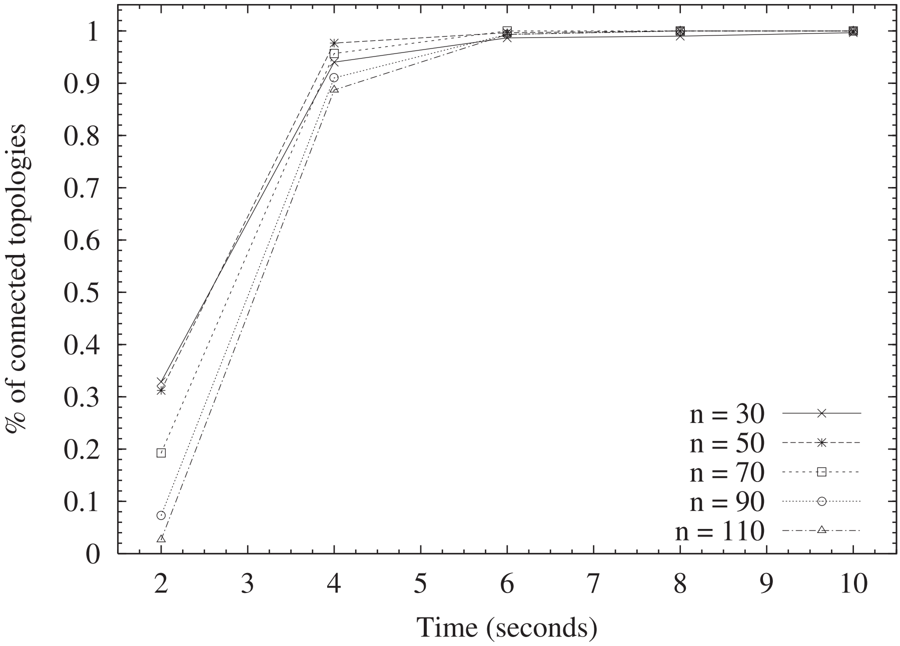


Figure 4.6. Percentage of connected BT topologies versus T_{disc}.

discovery phase, all neighboring nodes that have discovered each other exchange the list of the nodes they just discovered. This list exchange can be performed by executing the "pecking order" protocol as described above (Section 4.5) and leads to the construction at each node v of a set A_v of all nodes discovered by all v's neighbors that v did not discover.

Once the list exchange is finished, node v starts contacting the nodes in A_v to see whether they are nodes within its transmission range (i.e., undiscovered neighbors). To this purpose, a node v alternates for a predefined amount of time between page and page scan modes, attempting to discover the nodes in its A_v. More specifically, when in page mode, v attempts to page one after another (in round-robin fashion) of the nodes in A_v. Each time two nodes u and v discover each other, they remove each other from their sets A_u and A_v and exchange their lists of neighbors. This may lead to new nodes for u and v to be added to their sets A_u and A_v, that is, to new nodes to page. The length of this phase has to be carefully chosen so as to discover all the nodes in A_v that are actually in v's transmission range.

4.8 CONCLUSIONS

This chapter described solutions to the problem of scatternet formation, namely, to the problem of setting up multihop networks of Bluetooth devices. Solutions for the two major cases of when the devices are all in each others' transmission range and the more general case of multihop topologies have been illustrated. Two approaches for generating connected scatternets whose piconets have no more than seven slaves for multihop networks have been illustrated in detail. Observations and comments were given that described concerns arising while implementing scatternet formation protocols by following the current specifications (Version 1.1).

ACKNOWLEDGMENTS

The authors wish to thank Carla Fabiana Chiasserini, Imrich Chlamtac, Francesca Cuomo, Gabriele Mambrini, and Ivan Stojmenovic for valuable comments and useful discussions on the topics of this chapter.

REFERENCES

1. W. Kellerer, H.-J. Vögel, and K.-E. Steinberg, "A communication gateway for infrastructure independent 4G wireless access," *IEEE Communications Magazine, 40,* 3, pp. 126–131, March 2002.
2. http://www.bluetooth.com, *Specification of the Bluetooth System, Volume 1, Core,* Version 1.1, February 22, 2001.
3. B. N. Clark, C. J. Colburn, and D. S. Johnson, "Unit disk graphs," *Discrete Mathematics, 86,* 165–167, 1990.
4. G. Miklós, A. Rácz, Z. Turanyi, A. Valkó, and P. Johansson, "Performance aspects of Bluetooth scatternet formation," in *Proceedings of ACM MobiHoc 2000,* Boston, MA, August 11 2000.
5. P. Bhagwat and S. P. Rao, "On the characterization of Bluetooth scatternet topologies," http://www.winlab.rutgers.edu/pravin/bluetooth/index.html, 2000.
6. R. Guerin, E. Kim, and S. Sarkar, "Bluetooth technology: Key challenges and initial research," in *Proceedings of the SCS Conference on Networks and Distributed Systems, CNDS 2002,* San Antonio, TX, January 27–31 2002.
7. F. Cuomo and T. Melodia, "A general methodology and key metrics for scatternet formation in Bluetooth," in *Proceedings of IEEE Globecom 2002,* Taipei, Taiwan, 17–21 November 2002.
8. T. Salonidis, P. Bhagwat, L. Tassiulas, and R. LaMaire, "Distributed topology construction of Bluetooth personal area networks," in *Proceedings of the IEEE Infocom 2001,* Anchorage, AK, April 22–26 2001, pp. 1577–1586.
9. M. Ajmone Marsan, C. F. Chiasserini, A. Nucci, G. Carello, and L. De Giovanni, "Optimizing the topology of Bluetooth wireless personal area networks," in *Proceedings of IEEE Infocom 2002,* New York, 23–27 June 2002.
10. S. Baatz, C. Bieschke, M. Frank, P. Martini, C. Scholz, and C. Kühl, "Building efficient Bluetooth scatternet topologies from 1-factors," in *Proceedings of WOC 2002,* 2002.
11. C. Law, A. K. Mehta, and K.-Y. Siu, "A new Bluetooth scatternet formation protocol," *ACM/Kluwer Journal on Mobile Networks and Applications (MONET),* Special Issue on Mobile Ad Hoc Networks (A. Campbell, M. Conti and S. Giordano, eds.), *8,* 5, 485–498, October 2003.
12. G. Tan, A. Miu, J. Guttag, and H. Balakrishnan, "An efficient scatternet formation algorithm for dynamic environment," in *Proceedings of the IASTED Communications and Computer Networks (CCN),* Cambridge, MA, November 4–6 2002.
13. S. Basagni and C. Petrioli, "Multihop scatternet formation for Bluetooth networks," in *Proceedings of the 55th IEEE Semiannual Vehicular Technology Conference, VTC Spring 2002,* Birmingham, AL, May 6–9 2002, vol. 1, pp. 424–428.
14. C. Petrioli, S. Basagni, and I. Chlamtac, "Configuring BlueStars: Multihop scatternet formation for Bluetooth networks," *IEEE Transactions on Computers,* special issue on Wireless Internet (Y.-B. Lin and Y.-C. Tseng, eds.), *52,* 6, 779–790, June 2003.

15. G. Záruba, S. Basagni, and I. Chlamtac, "BlueTrees—Scatternet formation to enable Bluetooth-based personal area networks," in *Proceedings of the IEEE International Conference on Communications, ICC 2001,* Helsinki, Finland, June 11–14 2001, vol. 1, pp. 273–277.
16. Z. Wang, R. J. Thomas, and Z. Haas, "BlueNet—A new scatternet formation scheme," in *Proceedings of the 35th Hawaii International Conference on System Science (HICSS-35),* Big Island, Hawaii, January 7–10 2002.
17. I. Stojmenovic, "Dominating set based Bluetooth scatternet formation with localized maintenance," in *Proceedings of the Workshop on Advances in Parallel and Distributed Computational Models,* Fort Lauderdale, FL, April 2002.
18. X. Li and I. Stojmenovic, "Partial Delaunay triangulation and degree-limited localized Bluetooth scatternet formation," in *Proceedings of AD-HOC NetwOrks and Wireless (ADHOC-NOW),* Fields Institute, Toronto, Canada, September 20–21 2002.
19. C. Petrioli, S. Basagni, and I. Chlamtac, "BlueMesh: Degree-constrained multihop scatternet formation for Bluetooth networks," *ACM/Kluwer Journal on Special Topics in Mobile Networking and Applications (MONET),* Special Issue on Advances in Research of Wireless Personal Area Networking and Bluetooth Enabled Networks (G. Zaruba and P. Johansson, eds.), *9,* 1, 33–47, February 2004.
20. S. Basagni, "Distributed clustering for ad hoc networks," in *Proceedings of the 1999 International Symposium on Parallel Architectures, Algorithms, and Networks (I-SPAN'99),* A. Y. Zomaya, D. F. Hsu, O. Ibarra, S. Origuchi, D. Nassimi, and M. Palis eds., Perth/Fremantle, Australia, June 23–25 1999, pp. 310–315, IEEE Computer Society.
21. C. Petrioli and S. Basagni, "Degree-constrained multihop scatternet formation for Bluetooth networks," in *Proceedings of the IEEE Globecom 2002,* Taipei, Taiwan, R.O.C., November 17–21 2002, vol. 1, pp. 222–226.
22. I. Chlamtac and A. Faragó, "A new approach to the design and analysis of peer-to-peer mobile networks," *Wireless Networks, 5,* 3, 149–156, May 1999.
23. S. Basagni, R. Bruno, and C. Petrioli, "Performance evaluation of a new scatternet formation protocol for multi-hop Bluetooth networks," in *Proceedings of the 5th International Symposium on Personal Wireless Multimedia Communications, WPMC 2002,* Honolulu, Hawaii, October 27–30 2002, vol. 1, pp. 208–212.
24. S. Basagni, R. Bruno, G. Mamerini, and C. Petrioli, "Comparative performance evaluation of scatternet formation protocols for networks of Bluetooth devices," *Wireless Networks, 10,* 12, 197–213, March 2004.
25. A. C.-C. Yao, "On constructing minimum spanning trees in *k*-dimensional spaces and related problems," *SIAM Journal on Computing, 11,* 4, 721–736, November 1982.

CHAPTER 5

ANTENNA BEAMFORMING AND POWER CONTROL FOR AD HOC NETWORKS

RAM RAMANATHAN

5.1 INTRODUCTION

The physical layer underlying ad hoc networks has a number of parameters that can be controlled for improved performance. Such parameters include modulation, transmit power, spreading code, and antenna beams. By controlling these transceiver parameters adaptively and in an intelligent manner, one can increase the capacity of the system tremendously.

This chapter addresses the question: *how do we control the physical layer parameters for best performance?* We consider two parameters—transmit power and antenna beam direction—and present state-of-the-art methods for their control and their effect on performance. Although control of other parameters (such as modulation, coding, etc.) can also yield benefits, we shall focus on antennas and power control as they have been the most studied, perhaps because they are intuitively the easiest to exploit. Further, as we shall examine in considerable detail later, power control and beamforming are highly synergistic. It is therefore useful to study these parameters jointly, as this chapter does.

The benefits of antenna beamforming include reduced interference due to the narrower beamwidth, longer range due to higher signal-to-noise ratio (by virtue of higher gain and lesser multipath), and improved resistance to jamming. The benefits of power control include reduced interference and lower energy consumption. Beamforming and lower powers are also good for covertness [often referred to as low probability of detection (LPD)]. In sum, using antenna beamforming and power control enables higher capacity due to increased spatial reuse, lower latency, better connectivity, longer battery lifetime, and better security.

Mobile Ad Hoc Networking. Edited by Basagni, Conti, Giordano, and Stojmenovic.
ISBN 0-471-37313-3 © 2004 Institute of Electrical and Electronics Engineers, Inc.

Controlling antenna beamforming and transmit power judiciously is far from easy. Improper control may result in performance that is poorer than without such control. For instance, reducing the power too much may leave the network unconnected, or produce excessive delays. Further, directional transmissions introduce new hidden and exposed terminal problems that may cause a decrease in capacity if not addressed.

Thus, although the use of directional communications and power control appears to have potential, a number of questions need to be answered: What techniques are required for power and beamforming control? What are the tradeoffs involved in such control? Do existing network- and link-layer protocols have to be changed drastically? Does the performance improvement depend on the kind of antennas used, or the granularity of power control? Is this kind of control feasible in practice; has it been demonstrated? What kinds of performance improvements are possible, and what has been shown?

This chapter presents an overview of the work done toward answering these and other such questions. The goal is to provide the reader with an understanding of the problems encountered in the exploitation of beamforming antennas and power control, solution approaches to these problems, and their performance benefits. It is targeted toward the mobile ad hoc networking protocol researcher, providing her or him the necessary background, design tools, ideas, and insights for exploiting beamforming and power control at the medium-access and network layers.

We note that this is not a chapter on the physical layer. Rather, it is on how such higher layers as the medium-access control (MAC) and network layers can control the parameters of physical layer technologies. One can think of the details of the particular technology itself as a black box that offers “knobs” for control by higher layers. Such a control may be provided, architecturally, by way of application program interfaces (APIs) above the physical and the medium-access layers. Use of such APIs facilitates a clean way of controlling the transceiver parameters without layering violations.

This is also not a chapter on power *conservation*. Although battery savings may occur as a side benefit of interference-reducing mechanisms, that is not a focus. Rather, the focus is on using antenna and power control for increasing the capacity, reducing delay, and increasing the connectivity of ad hoc networks.

A typical ad hoc network needs a number of mechanisms at the link and network layers working in cohesion to provide data communications. The medium-access control (MAC) module provides distributed access to the channel, neighbor discovery is responsible for identifying nodes within one hop, routing determines routes to destinations, and the forwarding module uses this information for, say, hop-by-hop packet forwarding. The exact functions performed by these modules is obviously system- and protocol-specific. For instance, in some reactive (on-demand) protocols, there is no explicit neighbor discovery; this is implicitly done as part of routing.

Of these mechanisms, the MAC and the neighbor discovery are the most impacted with respect to antenna and power control. This is not surprising when you consider the fact that antenna beamforming and power control both most affect spatial reuse and communication range, which, respectively, are the focus of MAC and neighbor discovery. There are some opportunities for exploiting antenna beamforming and power control in some other mechanisms as well, for instance, in route discovery using directional transmissions, but the majority of the interesting issues (and therefore research) are in the MAC and neighbor discovery.

The majority of this chapter is thus devoted to a presentation of the problems in, and state-of-the-art solutions for, the four combinations shown in Table 5.1.

Table 5.1. The Four Different Areas Arising out of a Combination of Modules, and Physical Layer Parameters that are the Subject of this Chapter

	Beamforming	Power
Medium access	Directional MAC	Power-controlled MAC
Neighbor discovery	Antenna-based topology control	Power-based topology control

The rest of this chapter is organized as follows. We begin with a brief and very informal tutorial on beamforming antennas. Then, in Section 5.3, we discuss medium-access control, in particular, directional MAC, power-controlled MAC, and the benefits of combining the two controls. In Section 5.4, we discuss neighbor discovery, which results in a mechanism for topology control. We first discuss power-based topology control and then antenna-based topology control. Section 5.5 summarizes the chapter and overviews some open problems and interesting areas of research.

5.2 BEAMFORMING ANTENNAS

Of the two transceiver parameters that are the subject of this chapter, namely, transmit power and beamforming antennas, transmit power control is easily understood. Beamforming antennas, however, are a complex and intriguing subject that is not very well understood by the typical ad hoc neworking researcher. We therefore devote this section to a brief tutorial on beamforming antennas. This is not intended to cover all aspects of this technology, nor do we cover it precisely or formally. Rather, the idea is to give the basics in an informal and intuitive fashion to equip the reader unfamiliar with this topic with just enough knowledge to understand the remainder of this chapter. Readers familiar with beamforming antennas may skip this section. Readers wishing to explore this field in detail are referred to [1] and the citations therein.

5.2.1 Antenna Concepts

Radio antennas couple energy from one medium to another. An *isotropic antenna* radiates or receives energy equally well in all directions.[1] A *directional antenna* has certain preferred transmission and reception directions, that is, it transmits/receives more energy in one direction compared to the other.

The *gain* of an antenna is an important concept, and is used to quantify the directionality of an antenna. The gain of an antenna in a particular direction $\vec{d} = (\theta, \phi)$ is given [1] by

$$G(\vec{d}) = \eta \frac{U(\vec{d})}{U_{\text{ave}}} \tag{5.1}$$

where $U(\vec{d})$ is the power density in the direction $\vec{d}$, U_{ave} is the average power density over all directions, and η is the efficiency of the antenna that accounts for losses. Informally, gain measures the relative power in one direction compared to an omnidirectional anten-

[1] In reality, no antenna is perfectly omnidirectional, but we use this term to represent any antenna that is not intentionally directional.

na. Thus, the higher the gain, the more directional is the antenna. The *peak gain* is the maximum gain taken over all directions. When a single value is given for the gain of an antenna, it usually refers to the peak gain. Gain is often measured in unitless decibels (dBi), that is, $G_{\text{dBi}} = 10 \cdot \log_{10}(G_{\text{abs}})$, where G_{abs} is the absolute value of gain. An isotropic antenna has a gain of 0 dBi.

An *antenna pattern* is the specification of the gain values in each direction in space, sometimes depicted as projections on the azimuthal and elevation planes. It typically has a *main lobe* of peak gain and (smaller gain) *side lobes*. An example antenna pattern is shown in Figure 5.1. As is common practice, we use the word *beam* as a synonym for "lobe," especially when discussing antennas with multiple/controllable beams/lobes. A *null* is a direction of negative (in dBi) gain. For example, the pattern in Figure 5.1 has a null at 30 degrees.

We note that a larger gain in one direction necessarily results in a reduced gain in some other direction. Intuitively, one can think of an omnidirectional antenna's pattern as a ball of dough around the antenna. The volume of the ball represents the total power. Replacing this with a directional antenna causes the dough to be "squished" around so that some directions are pulled out (gain higher than 0 dB) and some are pushed in (gain lower than 0

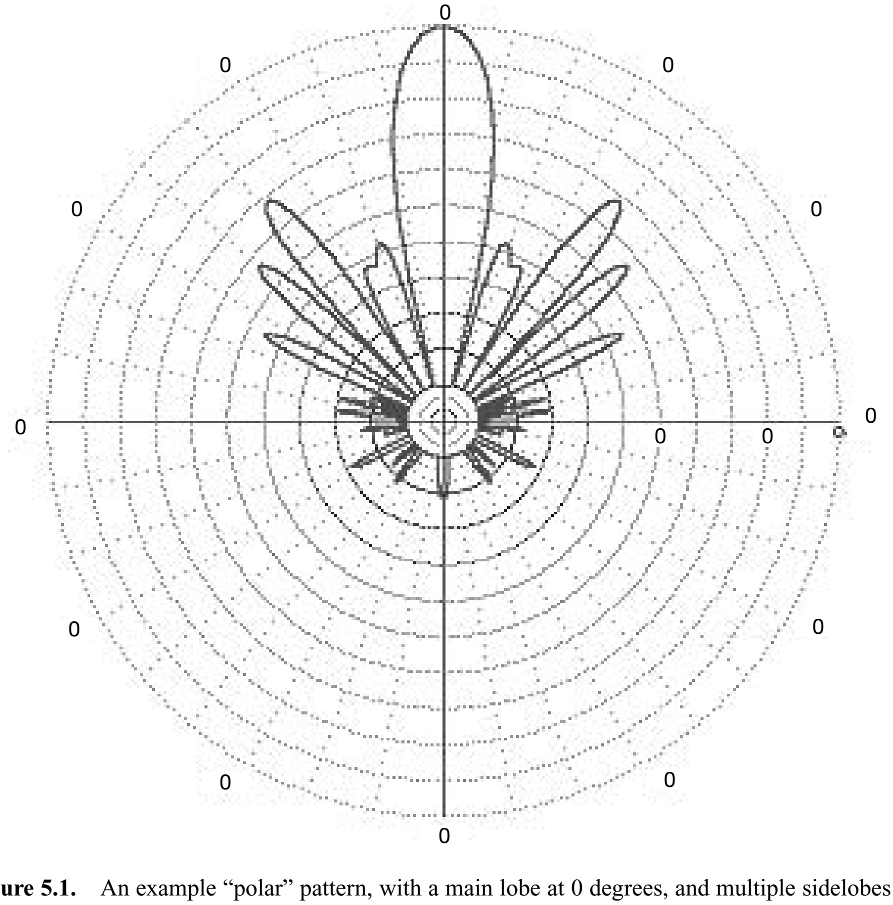


Figure 5.1. An example "polar" pattern, with a main lobe at 0 degrees, and multiple sidelobes of varying gains.

dB). But since the power emanating is the same (you only have so much dough), the lobes have to balance each other out, that is, preserve the law of conservation of power.

A related concept is the antenna *beamwidth*. Typically, this means the "3 dB beam width," which refers to the angle subtended by the two directions on either side of the direction of peak gain that are 3 dB down in gain. Gain and beamwidth are related. Typically, the more directional the antenna, the higher the gain and the smaller the beamwidth. However, two antennas with the same gain could have different beamwidths—for instance, the antenna with the smaller main lobe width may have more or larger sidelobes.

5.2.2 "Smart" Beamforming Antennas

The simplest way of improving the "intelligence" of antennas is to have multiple elements. The slight physical separation between elements results in signal *diversity* and can be used to counteract multipath effects. There are two well-known methods. In *switched diversity,* the system continually switches between elements so as to always use the element with the best signal. Although this reduces the negative effects of multipath fading, there is no increase in gain. In *diversity combining,* the phase error of multipath signals is corrected and the power combined to both reduce multipath and fading, as well as increase the gain.

The next step up in sophistication involves incorporating more control in the way the signals from multiple elements (the antenna *array*) are used to provide increased gain, more beams, and beam agility. Again, there are two main classes of techniques, as described below.

In *switched beam* systems, multiple fixed beams are formed by shifting the phase of each element's signal by a predetermined amount (this is done by a *beamforming network*), or simply by switching between several fixed directional antennas. The transceiver can then choose between one or more beams/antennas for transmitting or receiving. Although they provide increased spatial reuse, switched beam systems cannot track moving nodes, which therefore experience periods of lower gain as they move between beams.

In a *steered beam* system, the main lobe can be pointed virtually in any direction, often automatically using the received signal from the target and sophisticated "direction-of-arrival" techniques. One may distinguish between two kinds of steered beam systems—*dynamic phased arrays* that maximizes the gain toward the target, and *adaptive arrays* that additionally minimize the gain (produce *nulls*) toward interfering sources. The former allows *beam steering,* and the latter additionally provides *adaptive beamforming*.

In this chapter, we consider switched beam and steered beam antennas, jointly referred to as *beamforming antennas*.

5.2.3 Relevance for Ad Hoc Networks

When considering the use of beamforming antennas for ad hoc networks, a question is: *Aren't beamforming antennas too expensive and/or too big for ad hoc networks?* In this section, we argue that there do exist antenna techniques with suitable price and form-factor combinations.

Applications for ad hoc networking may be classified broadly into three categories: military, commercial outdoor, and commerical indoor, each with its own distinctive profile and able to accommodate different antenna technologies.

Military networks, which are by far the most prevalent application of *mobile* ad hoc networks, contain a significant number of large nodes (such as tanks, airplanes). The size of these platforms makes the form factor of most antennas quite irrelevant. Further, each platform by itself is so expensive that the cost of even the most sophisticated antenna is dwarfed by comparison. Thus, beamforming antennas are extremely relevant to military networks. An added bonus is that use of directional transmissions provides improved resistance to jammers and eavesdroppers.

Fixed ad hoc networks for commercial outdoor insfrastructure extend the reach of base stations using wireless repeaters organized into an ad hoc network. Packets are multihop routed through these repeater nodes with dynamic path selection. In some commercial approaches, the end-user terminals themselves serve as repeaters. Here, steered beam approaches may be too expensive. However, switched beams using inexpensive beamforming networks such as the Butler matrix [2] are easily manufactured using inexpensive hybrid couplers [1], making switched beamforming quite relevant.

The biggest deterrent to using beamforming antennas for networking small nodes such as PDAs and laptops within an indoor environment is the size. At 2.4 GHz and the typical half-wavelength element spacing, an eight-element cylindrical array would have a radius of about 8 cm, making it quite unwieldy. However, as the operating frequency continues to increase (the IEEE 802.11a is already working on wireless LANs in the 5 GHz band), the antenna sizes will shrink. At the 5.8 GHz ISM band, the eight-element cylindrical array will have a radius of only 3.3 cm, and at the 24 GHz ISM band, a mere 0.8 cm. Thus, the future looks bright for applying beamforming technology even to such applications.

Thus, while at first glance it may seem that ad hoc networks and beamforming antennas are not compatible, a more careful examination opens up a number of possibilities.

5.3 MEDIUM-ACCESS CONTROL

The goal of medium-access control (MAC) is to enable *efficient* sharing of the common wireless channel between nodes that need access to it. In order to be efficient, the MAC typically needs to employ *spatial reuse* of the channel, that is, it must provide as many simultaneous communications as possible.

Both transmit power and beamforming have an obvious and significant impact on spatial reuse. Reducing the transmit power reduces the circular range of interference and, thus, the number of interfered nodes. Directing the beam toward the intended receiver reduces energy in directions other than toward the receiver and, therefore, also reduces the number of interfered nodes.

Harnessing this potential, however, is nontrivial. Many MAC solutions tacitly assume homogeneous transmit power and/or omnidirectional transmissions. When these assumptions are violated, performance may deteriorate to below the performance when no control is used. We shall discuss these pitfalls in more detail later. For now, it suffices to say that techniques specifically targeted at supporting and exploiting power and beam control are required. Such techniques are the subject of this section.

Medium-access-control approaches may be broadly classified as either contention-based or contention-free. For ad hoc networks, the most commonly considered contention-based approach is CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance), and the most commonly considered contention-free approach is TDMA (Time Division Multiple Access). We shall survey adaptations of both CSMA/CA and

TDMA, but focusing more on CSMA/CA because, being the basis for the IEEE 802.11 standard, it has received far greater attention.

We begin with a treatment of directional medium access control, and then survey power-controlled MAC. Finally, we consider MAC solutions that exploit both beam and power control.

5.3.1 Directional MAC

We first consider CSMA/CA. Collision avoidance for ad hoc networks was first suggested by Karn [3], who proposed the MACA protocol. Many improvements on this were suggested in MACAW [4], FAMA [5], and others. The IEEE 802.11 standard Distributed Coordination Function (DCF) is based on CSMA/CA and is a good example of this approach. We describe it briefly below. Details can be found in [6, 7].

The IEEE 802.11 MAC Protocol. The IEEE 802.11 DCF used up to four frames for each data packet transfer. A sender first transmits a Request-to-Send (RTS), and the receiver responds with a Clear-to-Send (CTS). Then the sender sends the DATA and, finally, the receiver completes the transaction with an Acknowledgment (ACK). Both RTS and CTS contain the proposed duration of the data frame. Nodes located in the vicinity of the sender and the receiver that overhear one or both of the RTS/CTS store the duration information in a *network allocation vector* (NAV), and defer transmission for the proposed duration. This is called *virtual carrier sensing* (VCS).

The IEEE 802.11 protocol uses a backoff mechanism to resolve contentions. Before initiating transmission, the sender first waits for a short time to see if the channel is idle (this “inter-frame spacing” is different for different kinds of frames). Then, the sender chooses a random backoff interval from a range (0, CW) in which CW is the *contention window*. The sender then decrements the backoff counter once every “slot time.” When the backoff counter reaches 0, the sender transmits the frame. During this backoff stage, if a node senses the channel as busy, it freezes the backoff counter. When the channel is once again idle for a duration called DIFS (DCF Interframe Spacing), the node continues counting down from its previous (frozen) value.

If there is no CTS forthcoming from the receiver (due to collision, etc.), the sender doubles its CW, chooses a new backoff interval, and attempts retransmission. The contention window is doubled upon each such event until it reaches a maximum threshold. Upon successful delivery of a packet, the contention window is reset to CW.

We now consider the problem of adapting CSMA/CA and, in particular, 802.11 to the directional regime. An obvious solution to the problem is to do exactly as in IEEE 802.11, but simply send all of the RTS/CTS/DATA/ACK packets directionally. Unfortunately, this presents a number of problems. We discuss some of these problems below.

5.3.1.1 Problems with CSMA/CA When Beamforming Is Used. We present examples of four kinds of problems due to directionality of transmission/reception. The scenarios are taken from [8] and [9].

The first two examples are based on Figure 5.2. In the left-hand side of Figure 5.2, A wants to send a packet to B, and C wants to send a packet to D. The RTS from A is sent omnidirectionally or directionally to B, but is heard by C, and C is inhibited from sending even though it can do so without interfering with the transmission from A to B. We term this the *directional exposed terminal*.

In the right-hand side of Figure 5.2, suppose A is sending a packet to B after having ini-

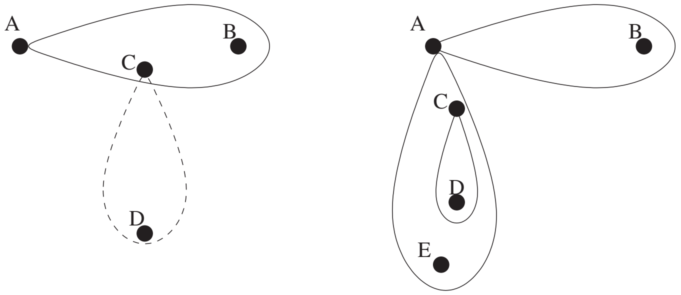


Figure 5.2. Examples of when traditional CSMA/CA is insufficient when beamforming is employed. In the example on the left, termed "directional exposed terminal," *C* is prevented from sending when it can. In the example on the right, termed "loss in channel state," there is a collision at D.

tiated an RTS–CTS exchange. Neither the RTS nor the CTS is heard by *C*, which proceeds to initiate a transmission to *D*. The RTS–CTS exchange between *C* and *D* is directional and not heard by *A*, which, after completing its transmission to *B*, now initiates a transmission to *E*. The RTS from *A* to *E* interferes with the data being received by *D* from *C*. We refer to this situation as *loss in channel state* by node *A*.

For the next two examples, we refer to Figure 5.3. The first of these examples is as follows. Assume all nodes in Figure 5.3 when idle, are receiving using an omnidirectional beam that has an azimuthal gain of G_o. Now, suppose *B* transmits a directional RTS to *F*, and *F* responds with a directional CTS. Suppose node A is far enough from node F so as to not be able to hear this CTS. *B* begins DATA transmission to F, both nodes pointing at each other with gain G_d. Because *B* sends directionally, and assuming very low gain in the opposite direction, *A* cannot sense this DATA transmission. While this is in progress, suppose *A* wants to send to *E*. When it sends the RTS directionally toward *E* (and it does so because it cannot sense anything), it interferes with the reception at *F*. This can happen because beams of both *F* and *A* are pointed toward each other. In other words, sender and receiver nodes with a combined gain of $G_d + G_o$ may be out of range of each other, but within range with a combined range of $2 \cdot G_d$ (note that G_d is by definition greater than G_o). We term this the *directional hidden terminal* problem.

For the final example, refer again to Figure 5.3. Suppose *B* initiates communication with *E* and starts sending a DATA to *E*. Further suppose that *C* is on a null for this transmission and so cannot hear these. While the *B–E* communication is ongoing, *C* wishes to send a DATA packet to *B*, and so sends an RTS to *B*. Since *B* is beamformed in the direction of *E*, it does not receive the RTS and so does not respond with a CTS. Node *C*, upon

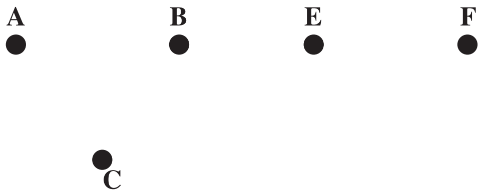


Figure 5.3. Figure to illustrate "directional hidden terminal" and "deafness" (refer to text)

not receiving the CTS, retransmits the RTS. This goes on until the RTS retranmission limit is exceeded. This wastes network capacity by sending unproductive control packets. Furthermore, *C* increases its backoff interval on each attempt, and thus, unfairness is introduced. This problem has been termed *deafness* in [9].

In addition to all this, there are some aspects of CSMA/CA protocol design that do not quite work when beamforming antennas are used. Consider the backoff scheme in IEEE 802.11, for example. This involves picking a random number of slots and counting down, freezing the count whenever the channel is busy. This is straightforward when there is only one beam that can be formed (omni), but is not so straightforward when beamforming is possible (steered or switched). While backing off, what beam should the node pick/form? Should it be omnidirectional, or should it continue to be beamformed in the direction of the intended transmission? If the former, the node misses the activity in the vicinity of the intended receiver. If the latter, then it may not be able to receive RTSs from nodes in other directions.

Last but not least, there is the issue of determining the direction in which the beam should be pointed to send to a target node or, in switched beams, which beam should be selected.

The examples outlined are by no means the only problems. However, they should have given the reader a flavor of the kinds of issues that researchers have grappled with in adapting CSMA/CA to beamformed transmissions. We now present a survey of research on solving these problems, under the informal umbrella of "directional CSMA/CA."

5.3.1.2 Directional CSMA/CA. The examples above indicate that the RTS/CTS handshake as used in traditional CSMA/CA is insufficient to overcome the new "directional" hidden- and exposed-terminal problems. In general, as the number of nodes hearing the RTS/CTS increases, the severity of the exposed-terminal problem increases, and that of the hidden-terminal problem decreases (the reader should examine the above examples again to grasp this). Thus, one idea is to attempt to find an optimum point by exploring the various combinations of omnidirectional/directional RTS/CTS.

The other, largely orthogonal, approach to the problem is to introduce mechanisms that explicitly try to reduce the exposed- or hidden-terminal problems. For instance, an exposed terminal may violate the protocol rules and transmit anyway to reuse the space.

These ideas, or a combination thereof, form the underlying rationale behind many of the schemes in the literature. A specific example is a simple scheme in which nodes send omnidirectional RTS/CTS, but nodes never honor the NAV—that is, they *always violate* the virtual carrier sensing—and the RTS/CTS is used merely to assist in the pointing of antennas for the subsequent DATA/ACK exchange and for power control. This was suggested as a simple baseline scheme in [8].

Ideas for determining the steering direction or beam selection include using the positions of the target node and reference node to derive the angle, using angle-of-arrival (AOA) facilities in steered (smart/array) antennas, etc. Directional MACs are generally indifferent as to which method is used, as long as the relative angle is obtained with reasonable accuracy. Note that it is the RTS direction that poses a challenging problem—the CTS/DATA/ACK can use position information placed in, or the angle-of-arrival of, the RTS/CTS/DATA (respectively) for determining the sending direction. To determine the RTS direction in the first place, one may use angle-of- arrival from overheard packets, obtain position information from overheard packets or (omnidirectional) beacons specifically used for this purpose, use piggybacked position information in routing control packets, or simply send the RTS omnidirectionally. Each of these techniques have their advantages and disadvantages. In the ensuing discussion, we focus more on the spatial reuse qualities of various schemes, treating the mechanics of direction determination as a largely orthogonal issue.

With this background in mind, let us now examine the ideas studied in the literature. We first consider the theme of omnidirectional/directional RTS/CTS, and then consider mechanisms for reducing collisions and increasing spatial reuse.

In [10], a MAC protocol is suggested for an ad hoc network in which each node has multiple directional antennas (functionally equivalent to a switched-beam system for protocol purposes) with a single transceiver. The idea is to execute the 802.11 protocol almost verbatim, but on a per-antenna basis. Thus, for instance, if a CTS is heard only on one antenna, an RTS may be sent out all other antennas except that one. Two schemes are described, both of which use directional DATA/ACK and omnidirectional CTS, but differ in the choice of how the RTS is sent—omnidirectionally or directionally. Simulations done with a mesh topology show, as expected, a throughput improvement over 802.11 with omnidirectional antennas. The relative performance of the two schemes is topology dependent. Although [10] was one of the early works on the problem and really kick-started this field, it makes a few assumptions that were later found to be unrealistic. For instance, it was assumed that the node can identify the antenna through which a packet was received, and that the omnidirectional antenna range is equal to the directional range.

The omnidirectional versus directional issue is examined more closely in [11]. Here, each node is assumed to have a switched-beam antenna using a beamforming matrix. Two schemes are compared. In both, CTS, DATA, and ACK are sent directionally. The difference is in the RTS—in one, called Di-RTS, it is sent directionally, and in the other, called omni-RTS it is sent omnidirectionally. Simulations reported show that the Di-RTS scheme outperforms the omni-RTS scheme significantly in all cases. The authors suggest that this is because the directional RTSs generate less interference. Another way of looking at this is that the exposed terminal problem affects throughput more than the hidden-terminal one.

Both RTS and CTS are sent omnidirectionally in the scheme in [26]. The authors consider an ad hoc network with steered beam antennas. In order to alleviate the exposed-terminal problem caused by the omnidirectionality of the RTS and CTS, the traditional backoff due to virtual carrier sensing is violated by using a shorter NAV for nodes not wanting to send to the source or destination of an ongoing communication. Thus, if a node C wanting to send to D hears an RTS from A to B, it will defer until the CTS from B to A is complete, and then will proceed to send an RTS to D even while A is sending DATA to B. Nodes A and B lock themselves into a tight directional transmit/receive link during data transfer and are largely immune to the communication between C and D. Power control is also used to reduce the signature of the DATA/ACK. Performance improvements over 802.11 of up to 130% in a 25 node grid and up to 260% in a 225 node grid are reported, even without power control. The power control aspect will be addressed in Section 5.3.2.1.

Omnidirectional RTS/CTS has the advantage that one need not know the position of the intended target. However, it cannot exploit the range advantage of directional antennas, that is, two nodes can talk with each other only when one of them beamforms to get the additional gain.

Unlike in omnidirectional networks, hearing an RTS/CTS does not always mean that it is necessary to defer. On the other hand, not deferring at all is obviously going to lead to collisions. Clearly, it is necessary to *selectively* defer depending upon the relative directions of the ongoing and intended transmissions. Such selective virtual carrier sensing using a *directional NAV* (or DNAV) was proposed in [12] and, independently, in [9]. The key idea is that if a node receives an RTS or CTS from a certain direction, then it needs to defer for only those transmissions that are in (and around) that direction. This is implemented by augmenting the NAV table entries with a direction field, and deferring only if the intended direction is within a threshold (for error margin) of that direction.

The scheme using such a DNAV in [12] improves network capacity by a factor of three to four over 802.11 for a 100 node network. This scheme uses "cached directions" based on angle-of-arrival information from overheard packets. This is used to send the RTS directionally. If the cache is empty, or if more than four RTS transmissions do not elicit a response, omnidirectional RTS is used. A hallmark of this work is the very comprehensive physical layer and antenna modeling that provides a high level of confidence in the results. The performance results in [9] indicate a throughput increase by a factor of about 2 over the traditional IEEE 802.11, but the antenna gains used were only 10 dBi. An interesting insight, pointed out in [9], is that if the flows are "aligned," directional transmissions perform much worse since packets from the same flow contend along the transmission direction.

There are a number of other works that consider directional CSMA/CA. In [13], signal strength information is used in lieu of position information to determine the angles. The novelty in [14] is in the use of an "Angle-SINR" table that keeps track of the communiation events and their directionality at any point in time. In [9], multihop RTSs are proposed for sending DATA to a receiver when both sender and receiver need to beamform toward each other for successful data transfer. Finally, a host of issues in directional CSMA/CA, including a comparison of switched and steered beams is presented in [8].

Are there any lessons to be learned from the research, and is there a convergence in thinking? Although this is still a very young field, some insights are emerging. First, there is a bunch of "low-hanging fruit" available for the taking—that is, even with simple modifications to CSMA/CA and moderate gain beams a capacity improvement of two to four is obtained. More sophisticated schemes should be able to increase this further. Second, as many packets should be sent directionally as possible (unless lack of position or other means forces one to use omnidirectional RTS). Third, a directional NAV and/or a short NAV is a good idea for exploiting the spatial reuse. Finally, and this will be discussed more in Section 5.3.2.1, augmenting beamforming with power control leads to a significant difference in performance.

5.3.1.3 Directional TDMA. An alternative approach to channel access in ad hoc networks is the *fixed,* or *contention-free* approach. The most studied manifestation of this approach in ad hoc networks is Time Division Multiple Access (TDMA). Although TDMA has not been studied as much as CSMA/CA (at least in recent times), there is no evidence that this is due to any inherent demerits[2] of TDMA. Rather, the relative simplicity of CSMA/CA and its adaptation by the IEEE 802.11 subcommittee is the likely cause of this relative imbalance in research.

In TDMA, time is divided into repeating *frames*. Each frame is divided into time *slots,*

[2] Some demerits are sensitivity to topological change and unsuitability for highly bursty traffic. However, these can be solved, and on the positive side it provides bounded delays and better utilization in general. TDMA versus CSMA/CA for ad hoc networks is an interesting topic, but beyond the focus of this chapter.

which are at least approximately synchronized. Transmissions start and end within slots. In a sufficiently spread out ad hoc network, slots can be reused by (adequately distant) nodes. Some researchers (e.g., [20]) refer to this specifically as *spatial reuse TDMA*. In the reminder of this discussion, we use TDMA as a synonymn for spatial reuse TDMA.

The use of TDMA in ad hoc networks with omnidirectional antennas has been extensively studied. This includes theoretical studies based on a graph-coloring paradigm [15, 16, 17], and distributed procedures [18, 19]. Slots can be assigned to nodes—called *broadcast scheduling* (which is more suitable for broadcast packets), or *link scheduling* (which is more suitable for unicast packets). In both cases, the *activations* (assignments of transmissions to a slot) must be made in a conflict-free manner: that is, the activations must adhere to a set of *constraints*. An example of a constraint is "Do not schedule concurrent transmissions at two nodes that are within two hops of each other." Another example is "With all scheduled transmissions active, the signal-to-interference-noise ratio (SINR) at all receivers must be above a certain threshold."

The use of beamforming antennas in TDMA poses problems that are completely different than with CSMA/CA. This is a consequence of the entirely different approach used for resolving contention—while CSMA/CA does this "on the fly" based on overheard control packets, TDMA does it apriori by coordinated decisions based on constraint information. In particular, the deafness and loss-in-state types of problems cease to be an issue. Although hidden and exposed terminals exist, the problem is different since this is resolved while setting up the schedule (and factored in as part of the "constraints").

With beamforming antennas, what changes is the *nature* of constraints for concurrent transmissions. For instance, consider Figure 5.4. In the figure, two links are constrained if they are adjacent, or if there is a line from a transmitter to a receiver. For instance, in Figure 5.4 (left) $A \rightarrow B$ and $D \rightarrow E$ are constrained, but $A \rightarrow B$ and $E \rightarrow F$ are not. Suppose all horizontal and vertical links need to be activated. The figure on the left shows interference constraints when omnidirectional tranmsissions are used and the figure on the right shows constraints when highly directional transmissions are used. With omnidirectional transmissions (left), many more links are constrained than with highly directional transmissions (right).

When scheduling, we need to ensure that there is no constraint line between a transmitter and its intended receiver. When omnidirectional antennas are used, only two links can be concurrently activated [Figure 5.4 (left)], namely $A \rightarrow B$ and $G \rightarrow H$. However, with directional communications [Figure 5.4 (right)], four links—$A \rightarrow B$, $D \rightarrow E$, $G \rightarrow H$, and $C \rightarrow F$—can be concurrently activated, giving a 100% increase in capacity.

As with directional CSMA/CA, obtaining the angle at which to transmit can be done using position information or angle of arrival. This may be done using specialized control packets or using a packet in the previous slot to get the updated position/angle of the target. Schemes discussed here are largely orthogonal to the specific mechanism for obtaining sending direction.

Given an existing omnidirectional TDMA design, and thus an extant vehicle for translating constraints into dynamic schedules, it appears incrementally less complex to accommodate directional antennas. What is needed is to determine the new set of constraints accurately. We discuss ideas to do this by describing two representative works [20, 21].

In [20], the authors study the performance of ad hoc networks with a TDMA MAC and two kinds of beamforming antennas—beam steering and adaptive beamforming. The algorithm used is a centralized one that uses two constraints: (1) in each slot, links are acti-

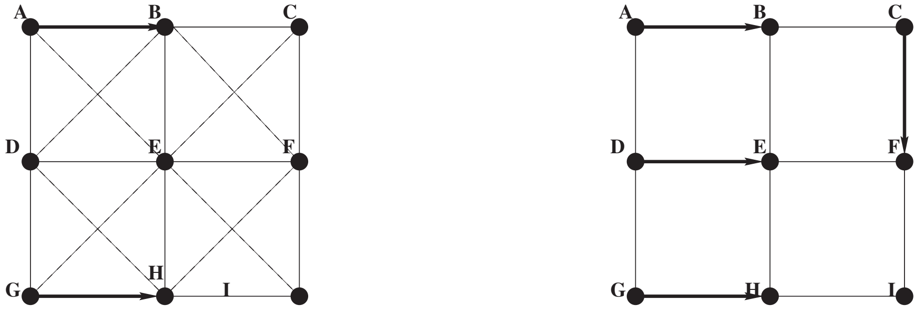


Figure 5.4. Constraints and activated links for a network with omnidirectional (left) and highly directional (right) beamforming. The reduction in constraints (indicated by solid or dashed lines) in the directional case allows four links to be simultaneously activated compared to two in the omnidirectional case. Activated links are shown as bold arrows.

vated such that the SINR is above a certain threshold; (2) a node can either transmit or receive one packet in a slot. The difference between directional and omnidirectional antennas is mostly found in (1). With beamforming, interference is significantly reduced at many nodes, thereby increasing SINR and allowing more simultaneous activations. Simulation results show that with beam steering for transmitting and adaptive beamforming for receiving, a capacity gain of about 980% over omnidirectional antennas is obtained.

Although this order-of-magnitude improvement is impressive, the algorithm used is centralized and is therefore ill-suited for ad hoc networks. In [21], a distributed algorithm is given that only uses two-hop information for scheduling, thereby making it scalable, yet implementable for mobile ad hoc networks. The beamforming antenna constraints are accommodated by using the concept of an *angular group*. An angular group corresponds to the coverage of a directional beam from a node and is used to determine conflicts. In particular, only activations resulting in disjoint angular groups are scheduled in the same slot. Each node is able to determine this using two-hop information exchanged by means of broadcast messages that use the omnidirectional mode on common control slots. The paper describes a simulation comparison with UxDMA [17], which describes a centralized heuristic for scheduling and shows significant performance gains.

5.3.2 Power-Controlled MAC

Traditional versions of CSMA/CA, including IEEE 802.11, FAMA, MACAW, and so on, assume that all nodes transmit all packets with the same power. However, this does not fully exploit the potential for spatial reuse. Spatial reuse using power control is possible at two levels: first, each node can be assigned a different maximum power that it must not exceed, but all packets from that node get sent with this power; second, within a given maximum power, a node modulates the individual powers of each packet—including control and DATA—so as to cause the least interference.

It is the second topic that is the subject of this section—the first problem of picking the right maximum power is addressed later in Section 5.4. We note that one can be done independently of the other and have their individual benefits.

We primarily consider CSMA/CA MACs here because, as with beamforming, most of the research has been done in this context. Further, we assume that all nodes only use omnidirectional transmissions. The combination of directional antennas and power control will be considered in Section 5.3.2.1.

A simple idea is to send the RTS and CTS at the maximum power and, using their received signal strengths, reduce the power of the DATA and ACK to the minimum required. However, this does not affect spatial reuse because the number of other nodes that defer based on the RTS and CTS is not reduced, and thus, there is no improvement in spectral reuse.[3] Thus, in order to obtain spatial reuse we need to send the RTS and CTS also at a reduced power. Ideally, this power should be just enough to reach the neighbors, and therefore might need to be different for different node pairs.

Unfortunately, CSMA/CA is inherently incompatible with such an approach. To see why, consider the example illustrated in Figure 5.5. Consider two nodes A and B that are close to each other, and suppose that A wishes to send to B. The CTS from B to A uses a (low) power, say P_1, which is less than the maximum possible power $P_{\max}$. Now, suppose another node C that cannot hear the RTS/CTS exchange between A and B wishes to send to a distant node D such that it has to use $P_{\max}$. If B is within range of C with $P_{\max}$, then the RTS and the DATA from C to D will collide with the DATA from A to B.

In other words, the RTS/CTS exchange does not prevent hidden-terminal problems when *heterogenous* power levels are used. In general, the control and data packets in a CSMA/CA regime must be transmitted with the *largest power* that any node can use [22] in order to guarantee collision-free floor acquisition. However, this brings us back to the original problem—sending only RTS/CTSs at the maximum power yields little additional benefit over no power control.

This dilemma has been addressed in [22, 23]. Both solutions make use of *busy tones* for virtual carrier sensing, and utilize the RTS/CTS more as a way of negotiating the correct power level for DATA. In a way, this is similar to the violation of the virtual carrier sensing described in the context of beamforming in Section 5.3.1.2. We describe the solution below, based on [22], called Power Controlled Multiple Access (PCMA).

In PCMA, collision avoidance is generalized to power control. Unlike the "on/off" model of conventional CSMA/CA, PCMA uses a "bounded power" model using two main mechanisms:

1. A Request-Power-to-Send (RPTS) and Acceptable-Power-to-Send (APTS) handshake used to determine the minimum power that will result in a succesful transmission. The RPTS/APTS transmission occurs in the *data channel*.
2. Each active receiver advertises its noise tolerance, given its current received signal and noise levels. This is done using *pulses* in the *busy tone channel*. The strength of the pulse indicates the tolerance to additional noise.

A sender node continuously monitors the busy tone channel to determine its power bound by measuring the maximum power received on the busy tone channel over a threshold time window. Then, using a backoff similar (but not identical) to traditional 802.11, the sender transmits an RPTS at a power level slightly below the bounded level.

Unlike the traditional RTS/CTS handshake, the RPTS/APTS handshake does not force

[3] In practice, depending upon the carrier sense threshold, one may see a slight improvement because nodes that backed off due to sensing the DATA/ACK may now be free to transmit.

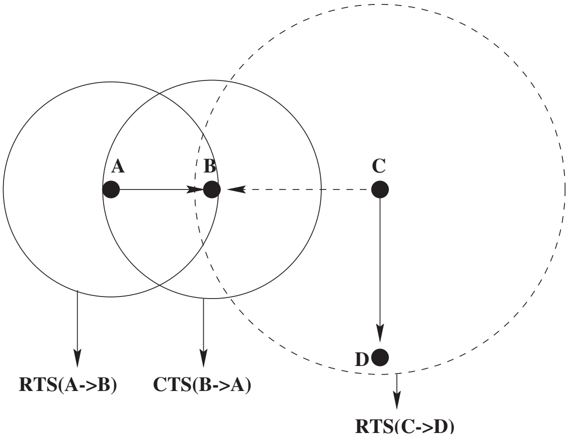


Figure 5.5. Heterogeneous transmit powers resulting from power control cause problems for CSMA/CA. Nodes *A* and *B* reduce power when sending RTS/CTS, which is not heard by *C*. When *C* transmits to a distant node *D*, it needs to use a high power, causing collision at *B*.

the hidden senders to back off. Rather, its utility is in calibrating the power level at which the DATA is to be sent. Specifically, based on its current noise level, the receiver computes the power *P* at which the DATA should be sent for it to be received successfully. This power *P* is included as part of the APTS packet, which is itself sent at reduced power based on the source's noise level information sent as part of the RPTS packet.

The solution in [23] also uses busy tones to coordinate power control, but the tones here are continuous. Two busy tones are used: a transmit busy tone, sent by an active transmitter, and a receive busy tone, sent by an active receiver. Different power levels are used for the tones by the transmitter and the receiver, and tuned appropriately.

Both [22] and [23] report significant performance gains from the respective methods using simulation. A throughput improvement by a factor of two over nonpower-controlled IEEE 802.11 is achieved using the mechanism in [22]. The approach of [23] is shown to provide approximately twice the peak channel utilization when compared to a nonpower-controlled dual busy tone technique, namely the one in [24].

Although the above solutions manage to avoid collisions, they require the use of a second channel for busy tones. This implies a *second transceiver* since the busy tones and transmission/reception may need to occur simultaneously. Thus, this is not possible with the off-the-shelf 802.11 wireless cards, and, indeed, one might argue that with two channels and two transceivers, nonpower-controlled 802.11 might itself exhibit close to factor-of-two performance gains. From a practical viewpoint, the problem that would be of most interest is a power-controlled CSMA/CA with a *single channel and transceiver*. As far as we know, this is an open problem. However, a trivial solution to this problem is to *not* address the collisions, resolving them by means of backoffs and retransmissions. Our experience, reported in [8] and in other unpublished research, is that one can still get substantial gains in performance.

Thus far, we have considered power control in the context of CSMA/CA. One of the few works that addresses power control in the TDMA context is [25]. The authors present a two-phase algorithm that searches for an admissible set of users in a slot, along with their transmission power. In the first phase, a scheduling algorithm is used to eliminate "strong" or "primary" constraints—for example, a node cannot simultaneously transmit and receive. In the second phase, a distributed algorithm is executed to control the admissible set of powers that could be used by nodes scheduled in phase 1. As in the case of TDMA with beamforming, power computation simply translates into a new set of constraints that need to be accommodated. In this case, the constraint is that the powers should be such that the SINR at every receiver must be greater than a given threshold.

5.3.2.1 Power-Controlled Directional MAC. Although beam and power control by themselves can improve spatial reuse considerably, it is when both are employed simultaneously that the full potential is realized. Figure 5.6 illustrates this very informally by comparing the four combinations—no power or beamforming control, only power control, only beamforming, and both. Ignoring a number of details such as sidelobes, one can take the ratio of the areas occupied by two schemes as a measure of the relative spatial reuse. Suppose that a node wants to send to a node that is at half the range of its maximum power (top left). With power control (top right), the relative area decreases by a factor of $\pi r^2/\pi(r/2)^2$ or four. With only beamforming and a beamwidth of 10 degrees (bottom left), energy is redirected but the same energy is emanated. Assuming, and because of, r^4 propagation, the energy interferes with less area than without power control or beamforming but still interferes significantly. Specifically, the range is $(360/10)^{1/4}$. This gives a factor-of-six improvement over no power control or beamforming, which is 50% better than with only power control. When both power and beamforming control are used (bottom right), the area occupied is approximately $(10/360) \cdot \pi(r/2)^2$, or a reduction in the area by a factor of 144!

In other words, the additional gain of the antenna in the preferred direction allows us to

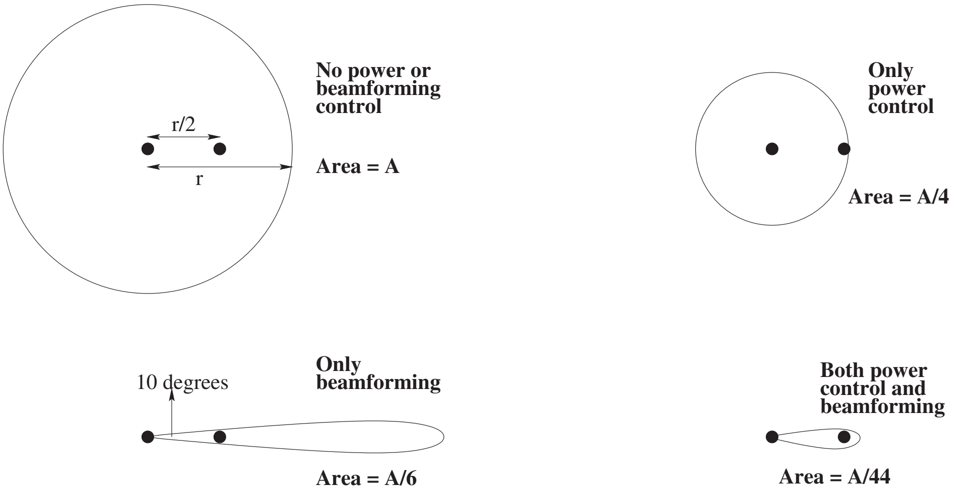


Figure 5.6. A rough comparison of relative interference reduction potential with power control, beamforming, and both together. Assuming a beamwidth of 10 degrees and r^4 propagation, and many simplifying assumptions, the area of interference is reduced by a factor of four with power control only, a factor of six with beamforming only, and a dramatic factor of 144 with both together.

reduce the power significantly. The savings depends upon the antenna gain—higher the gain, the more the savings. It is not surprising that the combination of power control and antenna gains is better than one or the other alone. What is surprising, and apparent from the above, is the *extent* of this difference. When combined, power control and beamforming are much more than the sum of the parts.

Figure 5.6 illustrates the relative *potential* for capacity enhancement. In order to harness this potential, protocols have to simultaneously control beamforming and power. One obviously cannot expect to match the theoretical potential, but the question of how much difference the incorporation of power control in beamforming makes merits attention. We now examine this further, and survey results on the comparative performance of beamforming with and without power control. We focus on two representative works, [8] and [26], targeting the question: given that beamforming is already done, what is the effect of doing power control in conjuction with it?

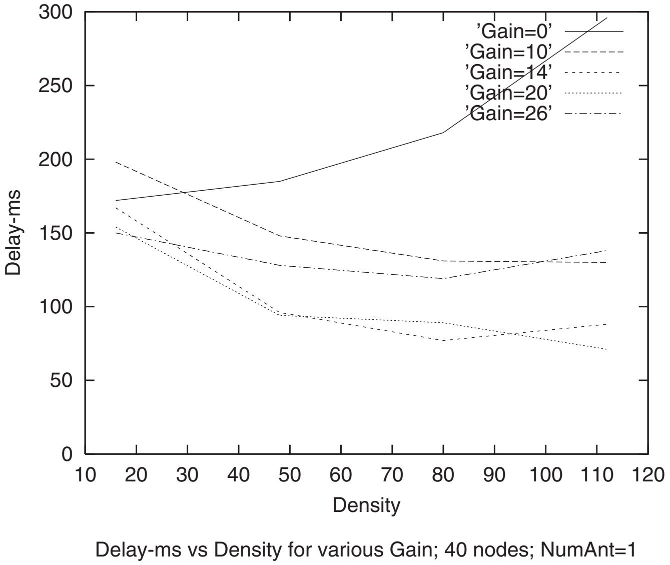
In [8], a simulation model of a 40 node static ad hoc network equipped with directional antennas, and CSMA/CA, MAC, and link-state routing are used to study the performance gains due to beamforming. The MAC protocol, called "Aggressive Collision Avoidance," sends RTS/CTS omnidirectionally, but virtual carrier sense is always violated; that is, the NAV is never honored. Idealized antenna patterns with varying gains are used—see [8] for details. Figure 5.7 shows the performance benefits with and without power control for various gains.

Because of the large packet buffers used, packets are delayed rather than dropped, and, therefore, the delay metric is a better indicator of performance here [8]. We note that without power control (Figure 5.7, top) there is a factor of two to three reduction in delay, whereas with power control (Figure 5.7, bottom), there is a factor of about 28 reduction in delay.

We now summarize the results from [26]. As mentioned earlier, in the protocol proposed here, RTS and CTS are sent using an omnidirectional antenna, and a short NAV is used to mitigate the exposed-terminal problem. In this context, the authors study two power control schemes—*global* and *local*. In both cases, power control is applied only to the DATA/ACK packets. In global power control (GPC), the DATA/ACK transmitter power is reduced to the same level for all nodes, to a value γP_t, where P_t is the transmitter power of RTS/CTS. In the local power control (LPC) scheme, the transmit power is set for each transmission so that the SNR across the link is a predetermined value. This is done by using the values of the received RTS/CTS power levels to compute how much power reduction is required.

The global and local power control schemes were compared with no power control (NPC) using a discrete event simulation on a 15 by 15 grid of 225 nodes. It was seen that the normalized system capacity increase over plain 802.11 was about 260% with NPC, about 475% with GPC, and about 525% with LPC. Thus, the addition of power control yields significant benefits. The paper concluded that reduction in power is a "key factor" in improving the capacity of an ad hoc network with smart antennas.

In sum, power control and beamforming are highly synergistic. Their use *together* far outperforms the sum of the individual gains achieved by use of one or the other by itself. Doing this, however, requires that we address the *union* of the MAC-layer issues presented in the exploitation of power and beamforming control. We believe that this should be a highly interesting and fruitful research area in the years to come.



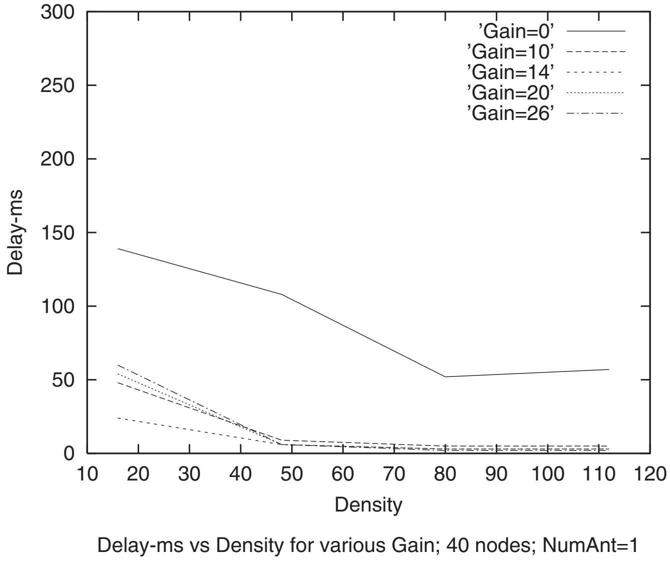


Figure 5.7. Performance of beamforming without (top) and with (bottom) power control: 40 node stationary ad hoc network with steered beams.

5.4 NEIGHBOR DISCOVERY: TOPOLOGY CONTROL

In the previous section, we studied the issues related to the *spatial reuse* of the spectrum. In this section, we consider the other dimension, namely, *communication range,* and, relatedly, *connectivity* control opportunities provided by antenna beam and power control.

The goal of neighbor discovery at each node is to determine the set of other nodes within direct communication range (within one hop). Neighbor discovery is an inherent part of most proactive protocols and uses a technique called *beaconing* to advertise itself and discover other nodes. In some reactive protocols, there is no explicit neighbor discovery. However, the process of building on-demand routes using route queries essentially discovers neighbors (that are in many cases cached for later use). One might therefore consider neighbor discovery as an implicit part of reactive protocols. Our description in this section is in the context of a proactive protocol, but we note that many of the ideas and issues are applicable in a modified form to reactive protocols as well.

The set of potential neighbors depends upon "uncontrollable" factors such as mobility, weather, noise, interference, and so on, as well as "controllable" ones such as transmit power and antenna direction. For instance, increasing the transmit power of beacons typically increases the number of neighbors. Similarly, depending upon whether the beaconing employs beamforming at one or both ends, different neighbors can be discovered.

The *topology* of the network is a union of S_i, where S_i is the set of links discovered by a node *i*. More generally, the topology of a network is the set of communication links used explicitly or implicitly by a routing mechanism [27]. Controlling the neighborhood of each node using power and beamforming obviously also controls the topology. This leads us to the notion of *topology control,* which is the problem of controlling the topology of the network to the desired form by changing the radio parameters; in this chapter, we only consider transmit power and antenna beam.

Why do we need topology control? Simply because the wrong topology can considerably reduce the effective capacity, increase the latency, and reduce the robustness of the network. For instance, if the topology is too sparse, there is a danger of loss in connectivity, and higher chances of congestion at a single node. On the other hand, a dense topology often implies reduced spatial reuse and increased battery consumption. Further, routing protocols might incur too much overhead in maintaining the topology, and this may overwhelm the routing process.

We observe that topology control can by accomplished in two ways:

1. Restrict the parameters to *physically* discover only some of the potential neighbors. We call this *physical topology control.*
2. Given a set of discovered neighbors, filter or refer only a subset of these neighbors to the routing process. We call this *routing topology control.*

The first use of the term topology control was as routing topology control [28] in the context of satellite networks. In [27], the term was reused, but meant physical topology control. Both techniques can be used for controlling the set of links seen by routing. The techniques are orthogonal and, thus, one can employ both physical and routing topology control to control the topology at two levels. In this chapter, we study only physical topology control. Unless explicitly mentioned, the term "topology control" will mean physical topology control.

In the next subsection, we shall discuss topology control based on power adjustment. The basic problem here is how to pick the right range for each node to balance connectivity, energy, and spatial reuse. Following that, we shall discuss topology control based on antenna beam pointing. The issue here is less about which nodes to pick[4] as how to form neighbors when one or both have to beamform toward the other.

[4] This is because, in this case, acquiring distant nodes as neighbors does not impact the spatial reuse nearly as much since transmissions are directional.

5.4.1 Power-Based Topology Control

The generalized problem of power-based[5] topology control (recall that we are considering only *physical* topology control in this chapter) is as follows [29, 27]: Given an ad hoc network, determine and adaptively adjust the transmit power of each node in the network so as to meet a given *minimization objective* and adhere to a given *connectivity constraint*.

The above is a generalized definition and subsumes a number of specific problems depending upon what is considered for the minimization objective and the connectivity constraint. Possibilities include:

- *Minimization objectives*. Maximum power, average power, maximum degree, average degree, maximum diameter, and so on.
- *Connectivity constraints*. Connectivity, biconnectivity, and so on.

An example of a specific topology control problem is one of dynamically adjusting transmit powers such that the maximum power used is minimized while keeping the network biconnected. This was first studied in [27]. Clearly, a number of other combinations are possible.

There is a further dimension to the problem relating to the dynamism of topology control. That is, just as many other distributed control problems, one can consider the *static* version of the problem or consider the *dynamic* version. In the static version, global topological information is available, the computation is "one-time" only, and it can use a *centralized* solution. In the dynamic version, only local information is available, the computation is continuous, and it typically requires a distributed algorithm. Although the dynamic version is more useful for mobile ad hoc networks, a study of the static version provides valuable insight, is useful to define the upper bound on performance, has some applicability to stationary ad hoc networks such as commercial mesh-based broadband wireless solutions, and is simple to understand.

In this subsection, we first examine the static version of the problem and examine some algorithmic issues. We then consider the dynamic version and survey some decentralized and distributed approaches.

5.4.1.1 Static Topology Control. We first introduce some terminology that will make the subsequent discussion less ambiguous.

First, an ad hoc network is represented as $M = (N, L)$, where N is a set of nodes and $L : N \longrightarrow (Z_0^+, Z_0^+)$ is a set of coordinates on the plane denoting the locations of the nodes.

The *least-power* function $\lambda(d)$ gives the minimum power needed to communicate over a distance of d. The function $\lambda(d)$ is dependent upon the propagation loss and the receiver sensitivity, and essentially maps from range to power (see [27] for details).

Using these, a graph-theoretic representation is used as follows. Given a multihop wireless network $M = (N, L)$, a transmit power function $\boldsymbol{p}$, and a least-power function λ, the *induced graph* is represented as $G = (V, E)$, where V is a set of vertices corresponding to nodes in N, and E is a set of undirected[6] edges such that $(u, v) \in E$ if and only if $p(u) \geq \lambda[d(u, v)]$, and $p(v) \geq \lambda[d(u, v)]$.

[5] In the remainder of this section, we shall omit "power based"; the term will be implied.

[6] An alternate and arguably superior representation would use directed edges to include unidirectional communication links. Note that we do not *assume* bidirectionality; we simply ignore unidirectional links. Using unidirectional links in an efficient manner requires sophisticated control protocols at several layers and is a subject of current research.

We use standard graph-theoretic terminology from [30]. In particular, a graph is said to be *k-vertex/edge connected* if and only if there are k vertex/edge-disjoint paths between every pair of vertices. Note that if a graph is k-vertex connected, then it is also k-edge connected, but the converse is not true. For this reason, and because vertex connectivity is important for resilience to node failures and hotspots, we shall consider only vertex connectivity. We shall omit the word "vertex" for brevity. Thus, if k is 1, the graph is *connected,* and if k is 2, it is *biconnected*. The *degree* of a vertex is the number of edges incident on that vertex. We only consider *undirected* graphs, that is, all edge relations on vertex pairs are symmetric.

Let us now consider a specific topology control problem, namely the one considered in [27]. In this problem, the constraint is 1-connectivity, and the objective is minimization of maximum power. This is stated formally below.

Definition 5.4.1 Problem: **Connected MinMax Power (C-MMP).** Given an $M = (N, L)$, and a least-power function λ, find a per-node minimal assignment of transmit powers $p : N \rightarrow Z^+$, such that the induced graph of (M, λ, p) is connected, and $\mathrm{MAX}_{u \in N}[p(u)]$ is a minimum.

An algorithm, called CONNECT, for this problem is given below. It is a simple "greedy" algorithm, similar to the minimum cost-spanning-tree algorithm. It works by iteratively merging connected components until there is just one. Initially, each node is its own component. Node pairs are selected in nondecreasing order of their mutual distance. If the nodes are in different components, then the transmit power of each is increased to be able to just reach the other. This is done until the network is connected. The description assumes for simplicity that network connectivity can be achieved without exceeding the maximum possible transmission powers. However, the algorithm can be easily modified to return a failure indication if this is not true.

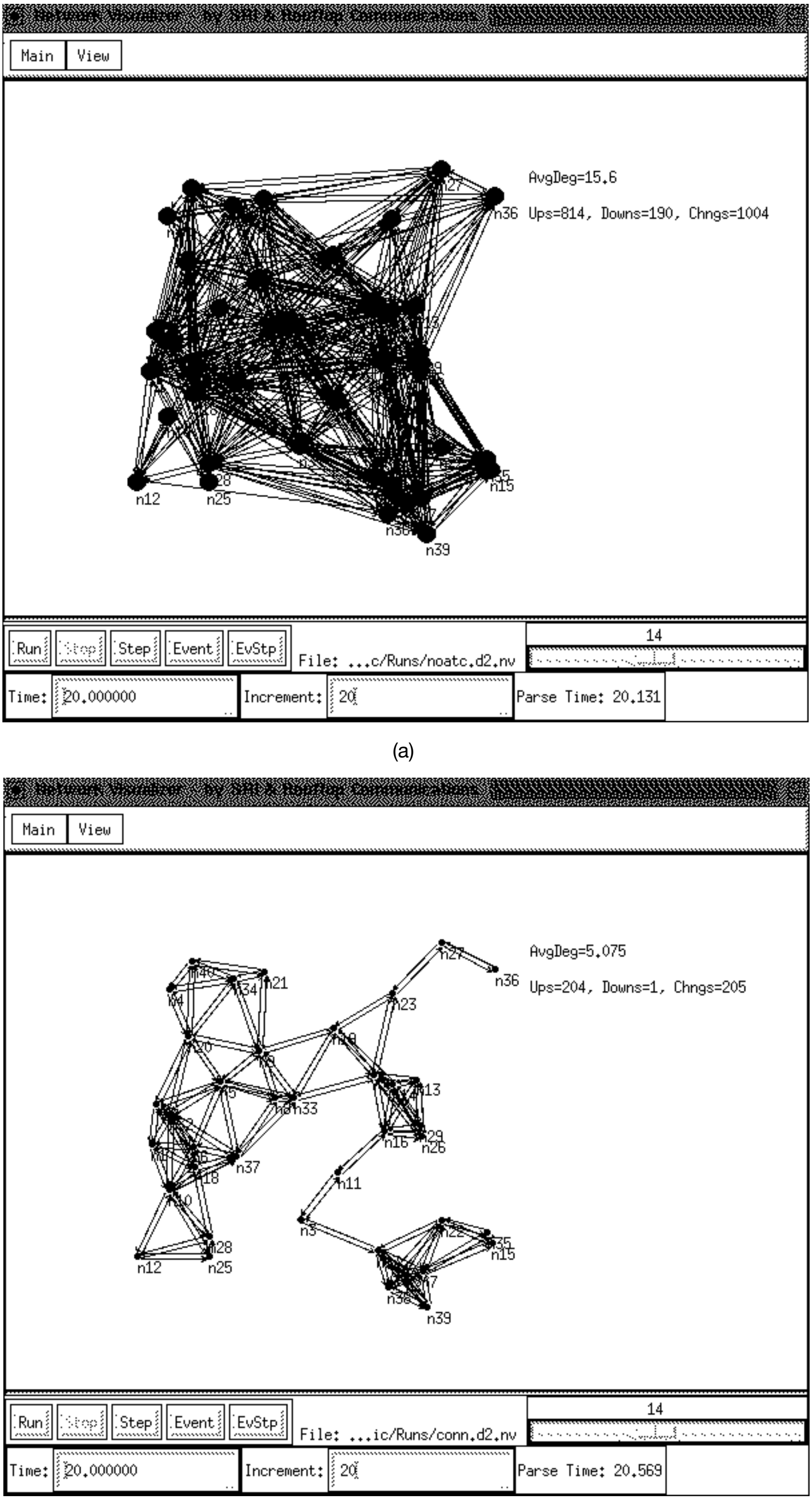
Algorithm CONNECT

Input: (1) Multihop wireless network $M = (N, L)$ (2) Least-power function λ

Output: Power levels p for each node that induces a connected graph

```
begin
1. sort node pairs in non-decreasing order of mutual distance
2. initialize $|N|$ clusters, one per node
3. for each $(u, v)$ in sorted order do
4.     if cluster(u) $\neq$ cluster(v)
5.         $p(u) = p(v) = \lambda(\text{distance}(u, v))$
6.         merge cluster(u) with cluster(v)
7.         if number of clusters is 1
           then end
8. perNodeMinimalize($M$, $\lambda$, $p$, 1)
end
```

Although this produces a minimum maximum power, it leaves some scope for reducing the powers of some nodes without affecting the connectivity. Line 8 in algorithm CONNECT is a procedure that exploits the presence of "side-effect edges" to minimize the power of each node, resulting in a reduced *average* power. The details of this procedure can be found in [27].



AvgDeg=15.6

Ups=814, Downs=190, Chngs=1004

File: ...c/Runs/noatc.d2.nv

14

Time: 20.000000

Increment: 20

Parse Time: 20.131

(a)

AvgDeg=5.075

Ups=204, Downs=1, Chngs=205

File: ...ic/Runs/conn.d2.nv

14

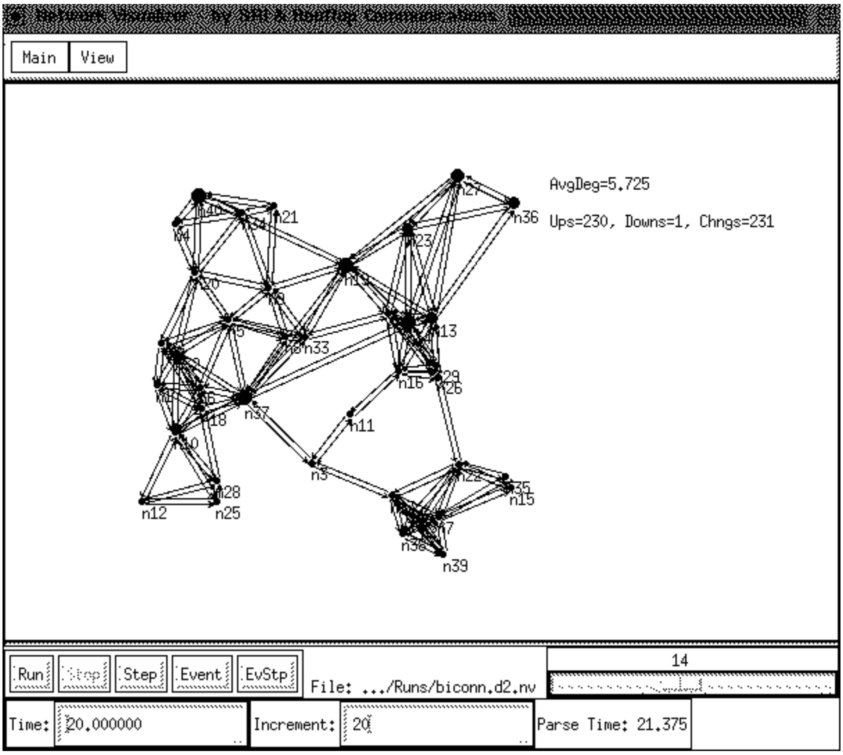
Time: 20.000000

Increment: 20

Parse Time: 20.569

(b)

Figure 5.8. (a) Uncontrolled network topology. (b) Connected topology using the C-MMP solution.



AvgDeg=5.725

Ups=230, Downs=1, Chngs=231

File: .../Runs/biconn.d2.nv

Time: 20.000000

Increment: 20

Parse Time: 21.375

(c)

Figure 5.8. (c) Biconnected topology using B-MMP solution.

This algorithm is provably optimal [27]. Algorithms for related problems—for example, the *Biconnected Minimum Maximum Power* (B-MMP), the *Connected Minimum Average Power* (C-MAP), and the *Connected Minimum Maximum Degree* (C-MMD)—are of a similar flavor. Algorithms for these problems are given in [29]. However, unlike the C-MMP, not all of them are amenable to optimal solutions. The algorithm for B-MMP is provably optimal [27], but C-MAP has been proven to be NP-complete in a variety of settings [31, 32], making it highly unlikely that there is a polynomial-time optimal solution.

The topology resulting from a solution for the C-MMP and the B-MMP problem on a network of 40 nodes spread out with a density of 2 nodes/sq mile is shown in Figure 5.8. The uncontrolled network (a) is too dense, whereas the connected network (b) has some congestion hot spots. The biconnected network (c) visually appears to provide the right balance, and, indeed, simulations in [27] show that it provides the best performance among the three.

The algorithmic aspects of topology control are studied thoroughly in [33]. There a general approach leading to an optimum polynomial–time algorithm is presented for minimizing maximum power for a class of graph properties called *monotone* properties. A property *P* is monotone if the property continues to hold even when the powers assigned to some nodes are increased while the powers assigned to other nodes remain unchanged. For example, *k connectivity* is monotone. Thus, the paper generalizes the results of [27] to hold for any *k-connectivity* constraint. They also give an approximation algorithm for the C-MAP problem that has a constant times–optimum performance guarantee.

Dynamic Topology Control. For dynamic networks, there are two approaches to doing topology control:

1. *Fully Distributed.* Nodes only know local information. All nodes execute the same (distributed) algorithm to produce the result.
2. *Decentralized.* All nodes have the global topology information through a flooded exchange, and run the same algorithm to produce the same result, after which each node uses the part of the result that pertains to itself. This is similar to the way the traditional link-state routing protocol works.

A "zero overhead" distributed algorithm called Local Information No Topology (LINT) is described in [27]. In LINT, a node is configured with three parameters—the "desired" node degree d_d, a high threshold on the node degree d_h, and a low threshold d_l. Periodically, the node checks the number of active neighbors (degree) in its neighbor table (built by the routing mechanism). If the degree is greater than d_h, the node reduces its operational power. If the degree is less than d_l, the node increases its operational power. If neither is true, no action is taken. The magnitude of the power change is a function of desired degree d_d and current degree d. In particular, the further apart d and d_d are, the greater is the magnitude of the change.

Although being extremely simple to implement, this algorithm does not guarantee connectivity. Other more sophisticated distributed algorithms that guarantee connectivity have been described, including those in [34, 35].

In [34], the concepts of a *relay region* and *enclosure* are used to select neighbors. A relay region for a node pair (i, r) is the region such that, for any node j in that region, it is more efficient for an $i \rightarrow j$ transmission to be relayed through r than to be sent directly. An enclosure for a node i is computed based on the relay regions of (i, n) for each neighbor n of i. Intuitively, the enclosure is a region beyond which it is not power-efficient to have neighbors. One of the key results is that if every node maintains communication links with the nodes in its enclosure, the network is strongly connected. The authors provide a distributed algorithm based on this that yields strong connectivity.

In [35], a cone-based distributed algorithm is described. A node continues to grow its power until its neighbor set is big enough such that, for any cone with angle α there is at least one neighbor, or until the node hits the maximum allowable power. They provide simulation results that outperform the algorithm in [34] in the scenarios studied.

The decentralized approach is inherently more overhead intensive, and less responsive to changes. On the other hand, it allows for direct execution of the static algorithms. A decentralized topology-control algorithm is described in [29]. The execution of that algorithm can be perceived as "punctuated equilibrium," with events happening at globally synchronized periodic intervals (only rough synchronization is required, as might be provided by a GPS clock). At these *topology reformation moments* (TRM), local state[7] is flooded, all nodes collect the individual local states to form the current snapshot of the global topology, execute a heuristic (e.g., algorithm CONNECT), and use the result for itself as the new power. Until the next TRM, this new set of powers is used.

[7] Two versions of "state" are described, one based on position, and another based on signal strength.

In [29], the decentralized solution to B-MMP (called GIFT, for "Global Information Full Topology") is compared with the Local Information No Topology (LINT) algorithm, and to the performance when no topology control is employed. We reproduce those results in Figure 5.9. All results reported here are for 60 nodes.

We note from Figure 5.9 that the decentralized algorithm yields the best throughput. The throughput at density 14 is about 11% better than that of LINT, and about 71% better than no topology control. However, the delay is also slightly higher, about 27% more than LINT. The increased delay is a result of being less densely connected (giving rise to larger number of hops between node pairs).

Although it appears wasteful, the decentralized approach exploits the fact that after the powers are set the first time, one can have considerable intervals between resets; that is, the TRMs can be fairly widely spaced apart. For instance, in the results discussed in the above paragraph, a 20 second interval was used. With this, the decentralized approach handily outperforms the LINT distributed algorithm for mobile networks, yet only uses about 0.5% of the capacity of a 2 Mbps transceiver. Thus, this is a simple yet scalable approach for at least nonhighly mobile ad hoc network applications.

5.4.2 Topology Control Using Beamforming Antennas

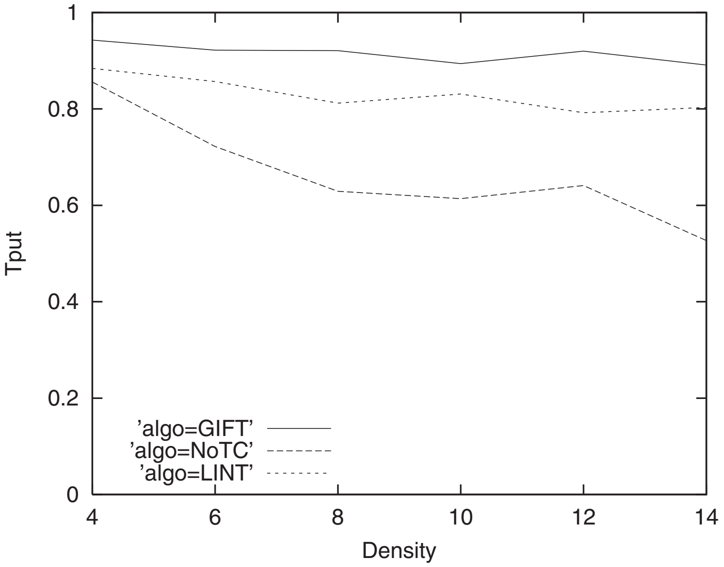
Transmitter beamforming provides additional gain in the direction that the packet is sent. Likewise, receiver beamforming provides additional gain in the direction from which a packet is received. These additional gains typically provide a significant increase in the range at which neighbors can be acquired. Moreover, which nodes can be neighbors depends upon whether none, one, or both of the nodes beamform. Orthogonally, other mechanisms for increasing the processing gain along the direction of communication also influence the ability to form neighbors.

Conventional neighbor discovery techniques such as those in [36, 37] assume the presence of omnidirectional antennas, and are not sufficient to enable discovery of neighbors with beamforming. Thus, one problem is: *How do we discover neighbors using beamforming?* While the problem is one of neighbor discovery, it is directly related to topology control because differing capabilities in discovering neighbors leads to differences in the way topology control can be effected.

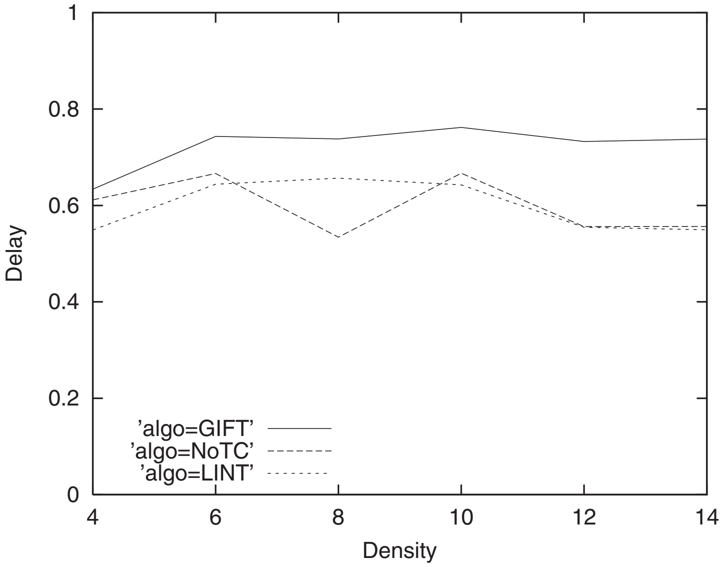
Once neighbors have been discovered, they have to be "maintained." Such maintenance typically uses periodic transmissions of beacons, nonreceipt of which indicates that the link has gone down. Unlike broadcast beaconing in omnidirectional networks, where a single beacon suffices for all directions, we now may need multiple beacons: one for each direction or neighbor.[8] Thus, we are led to the next problem: *How do we pick and choose the potential neighbors so as to get the desired topology?* This may be thought of as a dynamic degree-constrained network design problem. Very little work has been done on this, and, therefore, we only discuss the discovery problem in the remainder of this subsection.

A neighbor relationship may be specified as $b_s b_r$, where b_s is the beamform of the sender, and b_r is the beamform of the receiver. We consider two possibilities for b_s and b_r: omni (O) and directional (D). Thus, we have four kinds of neighbors—OO, DO, DD, and OD. This terminology was first used in [9]. The OO neighbors can be discovered us-

[8] Note that we cannot use omnidirectional beacons because their range may be considerably less and cannot reach the neighbors that have been acquired using beamforming.



Tput vs Density for various nbrMax; 60 nodes; Mobility=0.001, VC=N, LST=T



Delay vs Density for various nbrMax; 60 nodes; Mobility=0.001, VC=N, LST=T

Figure 5.9. A comparison of a decentralized algorithm (GIFT) (top) and a distributed algorithm (LINT) (bottom) with no topology control (NoTC) in a 60 node mobile ad hoc network.

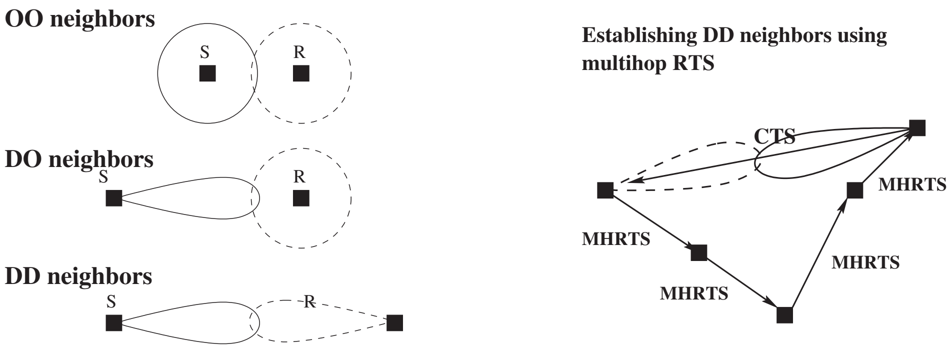


Figure 5.10. Left: OO, DO, and DD neighbors. Solid lines indicate transmitter beamforming, dashed lines indicate receiver beamforming. DO has longer range than OO, and DD has longer range than DO. Right: Illustration of multihop RTS for establishing DD neighbors.

ing traditional beaconing. Doing OD neighbor discovery is harder than doing DO, and does not yield additional range or other benefits. Thus, we shall examine only two types in greater detail here: DO and DD. Figure 5.10 (left) illustrates OO, DO, and DD neighbors.

We first consider DO discovery. A key issue here is to know which direction a node *A* must point to in order to send to a node *B*. There are two ways of doing this. If *B* is equipped with a smart antenna, it can eavesdrop on *A*'s transmissions and compute the angle of arrival (AOA). Another method is to use relayed position information. If a link-state protocol is used, such position information becomes automatically available if current position is included in each update. Alternatively, one may use efficient position dissemination techniques, as in [38, 39]. Using one's own position and the neighbor's position, the direction is easily computed. Assume that *A* knows the direction of *B* using one of these or other techniques. Then DO discovery is fairly simple: *A* beamforms towards *B* and sends a beacon. We assume that nodes are receiving in omnidirectional mode when not active. Thus, if the gain is sufficient, then *B* receives the beacon. The beacon contains *A*'s position. *B* uses that information to beamform toward *A* and send a beacon, enabling DO neighbor discovery.

DD discovery is more complicated, especially in a system that uses CSMA/CA at the MAC layer. In addition to the direction issue, which may be addressed in the same manner as for DO, a problem here is that the receiver must be beamformed in the direction of the sender at the precise time the beacon is sent. This is a problem not just for discovery, but for every single data packet transfer, although, in a TDMA system, scheduling the pointing could solve the data transfer problem.

Suppose, however, that the network is connected using links that are DO or OO. In that case, a *rendezvous* packet may be sent multihop from *A* to *B*. The rendezvous packet contains *A*'s position and the exact time at which *A* expects *B* to point toward *A* based on the position. Upon receipt of the rendezvous packet, and at the scheduled time, *A* and *B* can point to each other and try sending beacons to see if they can be DD neighbors.

In a CSMA/CA-based system, a *multihop RTS* may be used in place of a separate rendezvous packet. That is, when the beacon reaches the MAC layer, the MAC protocol determines that this is not a DO or OO neighbor and source-routes the RTS multihop to the receiver. During this time, *A* remains beamformed in the direction of *B*. Upon receipt of the RTS, which contains *A*'s position, *B* beamforms toward *A* and sends a CTS. If they can be DD neighbors, then the CTS will reach *A* and *A* can directly send the DATA (in this case the beacon). Such a multihop RTS scheme is described in [9], and further implementation details may be found there.

Clearly, this approach will only work if the network is connected using only DO and OO links. What if it is not? This is a hard problem. However, if another physical layer parameter, namely spreading gain, were controllable, one could trade data rate for increased processing gain (and, hence, range) just for the RTS, and use that to bootstrap the pointing.

What do DO and DD discovery give us? They essentially provide *range extension,* which, in turn, provides richer connectivity and a smaller average number of hops. Both of these are beneficial for the performance, but not under all circumstances. In the remainder of this section, we study some performance implications of range extension using beamforming antennas.

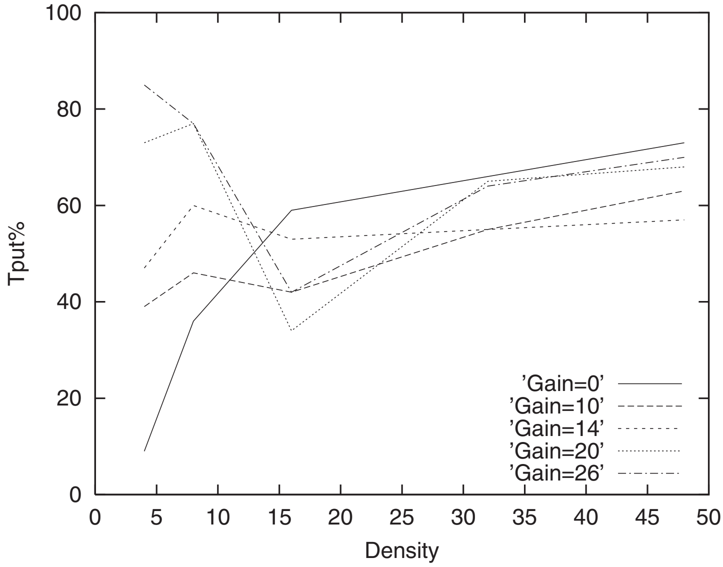
In [9] two protocols called DMAC and MMAC are compared. DMAC is a CSMA/CA-based protocol that uses a directional NAV table, as described in Section 5.3.1.2. DMAC only implements OO and DO modes. MMAC is an enhancement of DMAC with multihop RTS, which enables the DD mode. Thus, in DMAC, a packet may have to travel multiple hops, each of which involves an RTS/CTS/DATA/ACK exchange, whereas in MMAC, the packet may travel only one hop which involves a single MHRTS/CTS/DATA/ACK exchange. Using beamwidth of 45 degrees and a DD range of about 900 meters, compared to 250 meters for OO and a 25 node random network with random flows, the simulation results show that MMAC outperforms DMAC by a factor of up to 2.5. This clearly indicates the power of longer range transmissions to increase the capacity. MMAC also reduces the end-to-end delay by about 15%. The number is not very high because the queues saturate and packets are dropped (delay is only calculated for delivered packets). Further, the higher failure probability of multihop RTS in MMAC causes more timeouts and retransmissions, thereby offsetting some of the reduction in average packet latency.

In [8], the effect of range extension is studied using a model of switched beam antennas. A beacon is sent out on each of *K* beams. Since these beacons travel farther than omni beacons, longer-range neighbors and a richer topology are possible. The comparison is between OO- and DO-based topologies. The number of beams *K* is 12. A 40 node randomly placed static ad hoc network (see [8]) is used. The simulation results comparing the performance for different beam gains (including omnidirectional as gain of 0) using a realistic model in OPNET are in Figure 5.11.

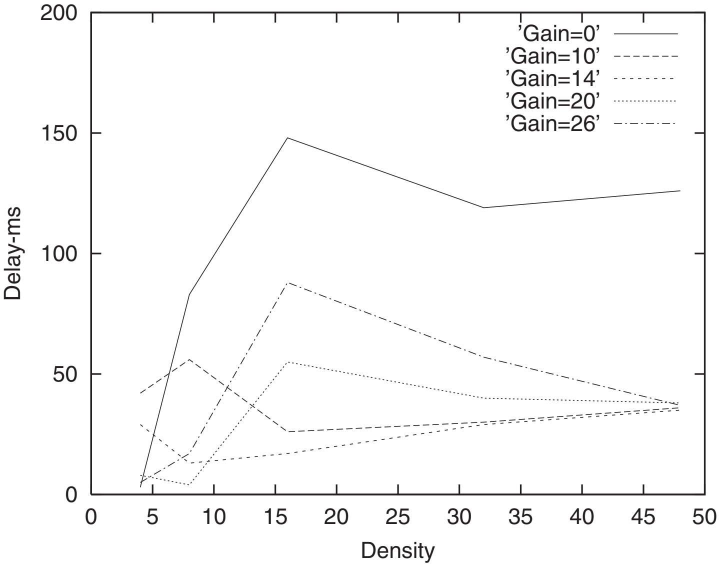
The relative performance appears to depend in a fairly complex manner on the density. Therefore, let us consider low, medium, and high densities separately.

At low densities, the throughput with beamforming antennas is far higher. For density 8, using a 20 dBi switched-beam antenna yields 118% better throughput and a factor of 20 reduction in delay.

For density 4 nodes/sq mile, a partitioned network results when omnidirectional antennas are used, but connected with beamforming antennas, as illustrated in Figure 5.12. For very sparse deployments, use of omnidirectional antennas leaves the network highly partitioned (a), whereas use of directional neighbor discovery with 10 dBi beams (b) and 20 dBi beams (c) provides good connectivity and commensurate performance. In such a scenario, the longer range provided by beamforming is simply indispensable to the connec-

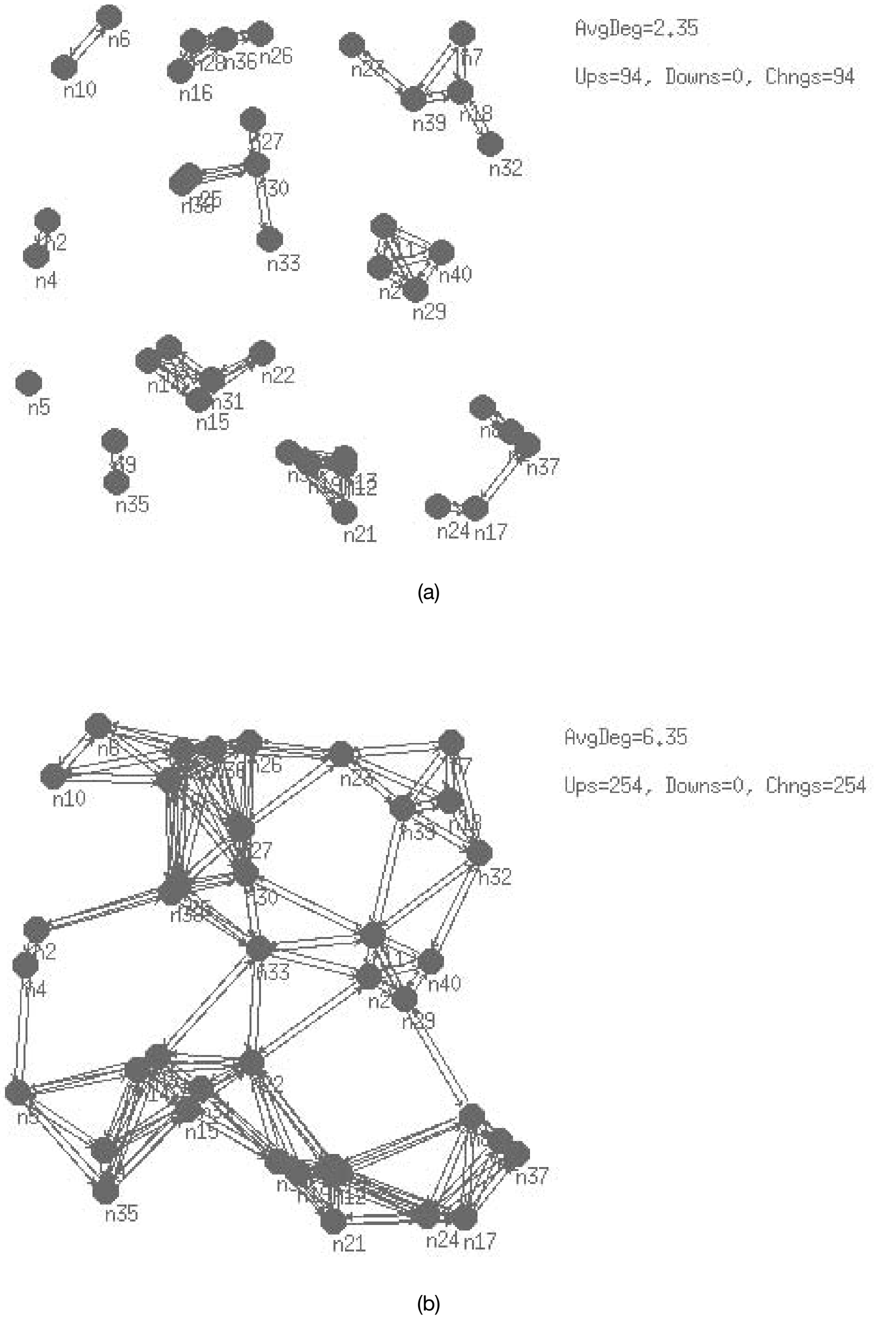


Tput% vs Density for various gains; 40 nodes; NumAnt=12



Delay-ms vs Density for various gains; 40 nodes; NumAnt=12

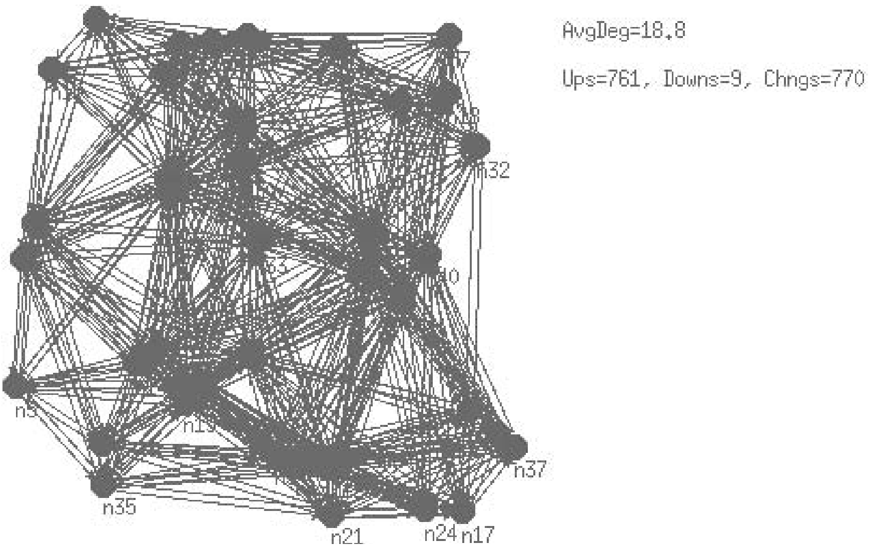
Figure 5.11. Comparision between using OO- and DO-based topologies. The "gain=0" indicates performance of OO topology. Performance of DO topologies with different values of transmitter beamforming gain are also shown. A 40 node stationary ad hoc network with switched beams and power control is modeled.



(a)

(b)

Figure 5.12. (a) Comparison of topologies resulting from omnidirectional antennas. (b) 10 dBi beams.



(c)

Figure 5.12. (c) 20 dBi beams.

tivity survivability of the network. We note that each link depicted is a directional link and, hence, unlike the omnidirectional case, a high average degree is not necessarily bad.

The performance drops at middle densities before rising again. This reflects a playing out of the interference versus range forces. That is, as density increases, the interference increases (which is bad) but the average number of hops decreases (which is good). At middle densities, the beneficial effects of the number of hops probably has not manifested itself, whereas the interference effects are dominant.

Now consider higher densities. The throughput is about the same or *worse with beamforming*. This is due to the fact that directional neighbor discovery tends to use the longer links by virtue of the shortest-path routing, which, in turn, causes more interference (recall the sidelobes). This bears out and extends to directional communications the conclusion in [40] that, all things being equal, one should use the smallest power (shortest links) that provides a connected network. This motivates the use of novel topology control and routing algorithms that use shorter links even when longer directional links exist. It also motivates cross-layer optimization.

An interesting question, and one that is unique to switched beams, is the dependence of the performance on the beamwidth and number of beams. Informally, one may think of the *coverage* of a switched beam system as $B \cdot N/360$, where B is the beamwidth of each beam in degrees and N is the number of beams. Reducing the beamwidth (and thus increasing the gain) has two counteracting effects: on the positive side, range is increased so farther neighbors are acquired into the topology; on the negative side, the coverage is decreased, and the number of "blind" spots increases (directions where there is little or no gain) increases, and even nearby neighbors that can be acquired with omnidirectional beams cannot now be acquired.

There has been no study of this trade-off. However, we note that that the optimal value of the coverage (with N constant) could well be less than 1. For instance, in Figure 5.11, consider density 8 and gain of 20 dBi (beamwidth of 20 degrees). The number of antennas used is 12, giving a coverage of 12 × 20/360, or 0.67. This does substantially better than omnidirectional (coverage of 1), and somewhat better than 10 dBi antenna (60 degree beamwidth, coverage of 2).

5.5 SUMMARY AND FUTURE DIRECTIONS

The control of transmit power and antenna beamforming by the medium-access and network layers offers a synergistic way of reducing interference while improving connectivity. To convert the potential into actual performance gains is a hard problem and requires radical modifications to existing mechanisms, or new mechanisms.

In this chapter, we considered mechanisms for the exploitation of antenna beamforming and power control for increased spatial reuse and richer connectivity with the goals of improved throughput, delay, and robustness in mind. Specifically, we considered the four combinations:

1. Medium access with beamforming antennas (Directional MAC).
2. Medium access with power control (Power-controlled MAC).
3. Neighbor discovery with beamforming antennas (Antenna-based topology control)
4. Neighbor discovery with power control (Power-based topology control).

Directional MAC in the context of CSMA/CA introduces difficult new problems such as directional hidden/exposed terminals, loss of NAV state, and so on. A number of new innovations such as the use of a short NAV, directional virtual carrier sensing, and so on. have been utilized to partially overcome these problems.

Power-controlled MAC allows a tighter packing of simultaneous transmissions within the network. However, new collision scenarios get introduced, an avoidance of which has thus far required additional capabilities such as busy tone.

Although power control and beamforming by themselves enable a considerable increase in the spatial reuse achivable by a channel-access mechanism, it is when both are used together that the full potential is achieved. This is intuitively apparent (see Figure 5.6), and borne out by simulations.

Neighbor discovery is the key mechanism that decides the routing topology of the network. This topology can be controlled using beamforming and power control. Power-based topology control offers an algorithmically rich set of problems. Although many of these problems are computationally intractable, distributed heuristics have been developed that provide a good balance between connectivity and spatial reuse.

There are other benefits of transmit-power control and beamforming that have not been addressed in this chapter. Chief among those is the capability for *covertness,* or, in military parlance, *low probability of detection* (LPD). Narrow power-controlled beams provide minimal leakage of energy so that intruders (eavesdroppers) can rarely intercept the packets. This is a physical layer security feature complementing other features such as spreading, encryption, and so on.

Although some advances have been made in the exploitation of beamforming antennas and power control for ad hoc networking, a lot more remains to be done. Some areas that offer exciting research opportunities include:

- Directional CSMA/CA. Current work still does not fully exploit the potential of beamforming antennas. Multicasting, multichannel operation, and QoS differentiation are some of the other issues in this context. Use of optimum power to further pack simultaneous transmissions is another area of interest. Finally, the problems of deafness and loss of state are among the least studied.
- Power-controlled CSMA/CA. The solutions for collision avoidance with heterogeneous powers rely on busy tones. Can one do this without using an additional channel?
- Neighbor discovery with beamforming. This is a very interesting and under-explored area. In particular, discovering neighbors when they are not connected by DO or OO paths is an important but difficult problem.
- Theoretical work on limits of ad hoc network capacity with power control and beamforming, for instance, an extension of [40].
- Exploitation of beamforming for other functions such as routing. An example is the use of directional antennas to reduce the query flooding overhead and simultaneously shorten the average hop count of routes in a reactive protocol.

Additionally, there are other physical-layer parameters such as modulation, spreading codes, and so on. that can also be exploited for improving ad hoc networking performance. A study of these in conjunction with beamforming and power control will open up several new frontiers in research.

REFERENCES

1. J. C. Liberti and T. S. Rappaport, *Smart Antennas for Wireless Communications,* Prentice-Hall PTR, 1999.
2. J. Butler and R. Lowe, “Beamforming Matrix Simplifies Design of Electronically Scanned Antennas,” *Electronic Design,* April 1961.
3. P. Karn, “MACA—A New Channel Access Method for Packet Radio,” in *Proceedings of ARRL/CCRL Amateur Radio 9th Computer Networking Conference,* September 1990.
4. V. Bhargavan, A. Demeers, S. Shenker, and L. Zhang, “MACAW—A Media Access Protocol for Wireless LANs,” in *Proceedings of ACM SIGCOMM ‘94,* September 1994.
5. C. L. Fullmer, J. J. Garcia-Luna-Aceves, “Floor Acquisition Multiple Access (FAMA) for Packet Radio Networks,” in *Proceedings of ACM SIGCOMM,* 1995, pp. 212–225.
6. IEEE Standards Department, *ANSI/IEEE Standard 802. 11—Wireless LAN,* IEEE Press, 1999.
7. M. S. Gast, *802. 11 Wireless Networks: The Definitive Guide,* O’Reilly and Associates, 2002.
8. R. Ramanathan, “On the Performance of Ad Hoc Networks with Beamforming Antennas,” in *Proceedings of ACM MobiHoc,* Long Beach, CA, October 2001.
9. R. R. Choudhury, X. Yang, R. Ramanathan, and N. Vaidya, “Using Directional Antennas for Medium Access Control in Ad Hoc Networks,” in *Proceedings of ACM MOBICOM,* Atlanta, Georgia, September 2002.
10. Y. B. Ko and N. H. Vaidya, “Medium Access Control Protocols Using Directional Antennas in Ad Hoc Networks,” in *Proceedings of IEEE INFOCOM,* March 2000.
11. M. Sanchez, T. Giles, and J. Zander, “CSMA/CA with Beam Forming Antennas in Multi-Hop Packet Radio,” in *Proceedings of the Swedish Workshop on Wireless Ad Hoc Networks,* March 2001.

12. M. Takai, J. Martin, A. Ren, and R. Bagrodia, "Directional Virtual Carrier Sensing for Directional Antennas in Mobile Ad Hoc Networks," in *Proceedings of ACM MOBIHOC,* Lausanne, Switzerland, June 2002.

13. A. Nasipuri, S. Ye, and R. E. Hiromoto, "A MAC Protocol for Mobile Ad Hoc Networks Using Directional Antennas," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC),* 2000.

14. S. Bandyopadhyay, K. Hasuike, and S. Horisawa, S. Taware, "An Adaptive MAC and Directional Routing Protocol for Ad Hoc Wireless Network Using ESPAR Antenna," in *Proceedings of ACM MOBIHOC,* Long Beach, California, October 2001.

15. E. Arikan, "Some Complexity Results About Packet Radio Networks," *IEEE Transactions on Inform. Theory, IT-30,* 910–918, July 1984.

16. N. Alon, A. Bar-Noy, N. Linial, and D. Peleg, " On the Complexity of Radio Communication," in *Proceedings of Twenty First Annual ACM Symposium on Theory of Computing,* pp. 274–285, 1989.

17. S. Ramanathan, "A Unified Framework and Algorithm for Channel Assignment in Wireless Networks," *Wireless Networks, 5,* 81–94, 1999.

18. I. Cidon and M. Sidi, "Distributed Assignment Algorithms for Multihop Radio Networks," *IEEE Transactions on Computers, 38,* 10, 1353–1361, Oct. 1989.

19. R. Ramaswami and K. K. Parhi, "Distributed Scheduling of Broadcasts in a Radio Network," in *Proceedings of the INFOCOM,* 1989.

20. K. Dyberg, L. Farman, F. Eklof, J. Gronkvist, U. Sterner, and J. Rantakokko, "On the Performance of Antenna Arrays in Spatial Reuse TDMA Ad Hoc Networks," in *Proceedings of IEEE MILCOM,* Anaheim, California, October 2002.

21. L. Bao, and J. J. Garcia-Luna-Aceves, "Transmission Scheduling in Ad Hoc Networks with Directional Antennas," in *Proceedings of ACM MOBICOM,* Atlanta, Georgia, September 2002.

22. J. P. Monks, V. Bhargavan, and W. W. Hwu, "A Power Controlled Multiple Access Protocol for Wireless Packet Networks," in *Proceedings of IEEE INFOCOM,* Anchorage, Alaska, April 2001.

23. S.-L. Wu, Y.-C. Tseng, and J.-P. Sheu, "Intelligent Medium Access for Mobile Ad Hoc Networks with Busy Tones and Power Control," *IEEE Journal on Selected Areas in Communications, 18,* 9, September 2000.

24. J. Deng and Z. J. Haas, "Dual Busy Tone Multiple Access (DBTMA): A New Medium Access Control for Packet Radio Networks," in *Proceedings ICUPC,* October 1998.

25. T. ElBatt and A. Ephremides, "Joint Scheduling and Power Control for Wireless Ad-Hoc Networks," in *Proceedings of IEEE INFOCOM,* New York, June 2002.

26. N. S. Fahmy, T. D. Todd, and V. Kezys, "Ad Hoc Networks with Smart Antennas Using IEEE 802. 11-Based Protocols," in *Proceedings of IEEE ICC,* 2002.

27. R. Ramanathan and R. Hain, "Topology Control of Multihop Radio Networks using Transmit Power Adjustment," in *Proceedings of IEEE INFOCOM,* Tel Aviv, Israel, 2000.

28. N. Shacham, "Protocols for Multi-Satellite Networks," in *Proceedings of IEEE MILCOM,* 1988.

29. R. Ramanathan, "Making Ad Hoc Networks Density Adaptive," in *Proceedings of IEEE MILCOM,* Vienna, Virginia, October 2001.

30. F. Harary, *Graph Theory,* Addison-Wesley, 1972.

31. W. Chen and N. Huang, "The Strongly Connecting Problem on Multihop Packet Radiio Networks," *IEEE Transactions on Communications, 37,* 3, March 1989.

32. A. E. F. Clementi, P. Penna, and R. Silvestri, "Hardness Results for the Power Range Assignment Problem in Packet Radio Networks," in *Proceedings of 3rd International Workshop on*

Randomization and Approximation in Computer Science (APPROX 1999), Lecture Notes in Computer Science, Vol. 1671, Springer-Verlag, 1999, pp. 195–208.

33. E. L. Lloyd, R. Liu, M. V. Marathe, R. Ramanathan, and S. S. Ravi, "Algorithmic Aspects of Topology Control Problems for Ad Hoc Networks," in *Proceedings of ACM MOBIHOC,* Lausanne, Switzerland, June 2002.
34. V. Rodoplu and T. H. Meng, "Minimum Energy Mobile Wireless Networks," *IEEE Journal on Selected Areas in Communications, 17,* 8, 1333–1344, August 1999.
35. R. Wattenhofer, L. Li, P. Bahl, and Y-M. Wang, "Distributed Topology Control for Power Efficient Operation in Multihop Wireless Ad Hoc Networks," in *Proceedings of IEEE INFOCOM,* Anchorage, Alaska, April 2001.
36. P. Jacquet, P. Muhlethaler, and A. Quayyum, "Optimized Link State Routing Protocol," IETF MANET Working Group Internet-Draft, Work in Progress.
37. B. Bellur and R. Ogier, "A Reliable, Efficient Topology Broadcast Algorithm for Dynamic Networks," in *Proceedings of IEEE INFOCOM,* 1999.
38. Y. B. Ko and N. H. Vaidya. "Location-Aided Routing (LAR) in Mobile Ad Hoc Networks," in *Proceedings of ACM/IEEE MOBICOM,* Dallas, TX, October 25–30, 1998.
39. S. Basagni, I. Chlamtac, V. R. Syrotiuk, and B. A. Woodward. "A Distance Routing Effect Algorithm for Mobility (DREAM)," in *Proceedings of ACM/IEEE MOBICOM,* Dallas, TX, October 25–30, 1998.
40. P. Gupta and P. R. Kumar, "The Capacity of Wireless Networks," *IEEE Transactions on Information Theory, IT-46,* 2, 388–404, March 2000.

CHAPTER 6

TOPOLOGY CONTROL IN WIRELESS AD HOC NETWORKS

XIANG-YANG LI

6.1 INTRODUCTION

Recent years have seen a great amount of research in wireless networks, especially ad hoc wireless networks due to their potential applications in various situations such as battlefield, emergency relief, and so on. There are no wired infrastructures or cellular networks in *ad hoc* wireless network. In this chapter, we assume that each wireless node has an omnidirectional antenna and a single transmission of a node can be received by *any* node within its vicinity, which we assume is a disk centered at this node. We also discuss specifically the topology control when directional antennas are used. Each mobile node has a transmission range. Node v can receive the signal from node u if node v is within the transmission range of the sender u. Otherwise, they communicate through multihop wireless links by using intermediate nodes to relay messages. Consequently, each node in the wireless network also acts as a router, forwarding data packets for other nodes. In addition, we assume that each node has a low-power Global Position System (GPS) receiver, which provides the position information of the node itself. If GPS is not available, the distance between neighboring nodes can be estimated on the basis of incoming signal strengths and the direction of arrival. Relative coordinates of neighboring nodes can be obtained by exchanging such information between neighbors [1].

It is common to separate the network design problem from the management and control of the network in the communication network literature. The separation is very convenient and helps to significantly simplify these two tasks, which are already very complex on their own. Nevertheless, there is a price to be paid for this modularity, as the decisions made at the network-design phase may strongly affect the network management and con-

Mobile Ad Hoc Networking. Edited by Basagni, Conti, Giordano, and Stojmenovic.
ISBN 0-471-37313-3 © 2004 Institute of Electrical and Electronics Engineers, Inc.

trol phase. In particular, if the issue of designing efficient routing schemes is not taken into account by the network designers, then the constructed network might not be suited for supporting a good routing scheme. For example, a backbone like network topology is more suitable for a hierarchical routing method than a flat network topology.

Wireless ad hoc networks need some special treatment as they intrinsically have their own special characteristics and some unavoidable limitations compared with wired networks. For example, wireless nodes are often powered by batteries only, and they often have limited memories. A transmission by a wireless device is often received by many nodes within its vicinity, which possibly causes signal interferences at these neighboring nodes. On the other hand, we can also utilize this property to save the communications needed to send some information. Unlike most traditional static communication devices, the wireless devices often are moving during the communication. Therefore, it is more challenging to design a network topology for wireless ad hoc networks that is suitable for designing an efficient routing scheme to save energy and storage memory consumption than for the traditional wired networks.

To simplify the question so we can derive some meaningful understanding of wireless ad hoc networks, we assume that the wireless nodes are quasistatic for a period of time. Then, in technical terms, the question we deal with is whether it is possible (and if possible, then how) to design a network that is a subgraph of the unit disk graph, such that it ensures both attractive network features such as bounded node degree, low stretch factor, and linear number of links, and attractive routing schemes such as localized routing with guaranteed performance.

Unlike the wired networks that typically have fixed network topologies, each node in a wireless network can potentially change the network topology by adjusting its transmission range and/or selecting specific nodes to forward its messages, thus controlling its set of neighbors. The primary goal of topology control in wireless networks is to maintain network connectivity, optimize network lifetime and throughput, and make it possible to design power-efficient routing. Not every connected subgraph of the unit disk graph plays the same important role in network design. One of the perceptible requirements of topology control is to construct a subgraph such that the shortest path connecting any two nodes in the subgraph is not much longer than the shortest path connecting them in the original unit disk graph. This aspect of path quality is captured by the *stretch factor* of the subgraph. A subgraph with constant stretch factor is often called a *spanner,* and a spanner is called a *sparse spanner* if it has only a linear number of links. In this chapter we review and study how to construct a sparse network topology efficiently for a set of static wireless nodes.

Restricting the size of the network has been found to be extremely important in reducing the amount of routing information. The notion of establishing a subset of nodes that perform the routing has been proposed in many routing algorithms [2–5]. These methods often construct a virtual backbone by using the connected dominating set [6–8], which is often constructed from a dominating set or a maximal independent set. For a full review of the state of the art in constructing the backbone, see Li [9].

The other imperative requirement for network topology control in wireless ad hoc networks is the fault tolerance. To guarantee a good fault tolerance, the underlying network structure must be k-connected for some $k > 1$, i.e., given any pair of wireless nodes, there need to be at least k disjoint paths to connect them. By setting the transmission range sufficiently large, the induced unit disk graph will be k-connected without doubt. Since energy conservation is important to increase the life of the wireless device, the question is how to find the minimum transmission range such that the induced unit disk graph is multiply connected.

Many routing algorithms [3, 5, 11–14] have been proposed recently for wireless ad hoc networks. The routing protocols proposed may be categorized as table-driven protocols or demand-driven protocols. A good survey may be found in [15]. Route discovery can be very expensive, 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 scarce memory of wireless network nodes. The geometric nature of the multihop ad-hoc wireless networks allows a promising solution: localized routing protocols.

Localized routing does not require the nodes to maintain routing tables, a distinct advantage given the scarce storage resources and the relatively low computational power available to the wireless nodes. More importantly, given the numerous changes in topology expected in ad-hoc networks, no recomputation of the routing tables is needed and, therefore, we expect a significant reduction in overhead. Thus, localized routing is scalable. Localized routing is also uniform, in the sense that all the nodes execute the same protocol when deciding to which other node to forward a packet.

But localized routing is challenging to design, as even guaranteeing the successful arrival at the destination of the packet is a nontrivial task. This task was successfully solved by Bose et al. [16] (see also [17]). Mauve et al. [18] conducted an excellent survey of position-based localized routing protocols.

6.1.1 Organization

The rest of the chapter is organized as follows. In Section 6.2, we review in detail the geometry structures that are suitable for topology control in wireless ad hoc networks, especially the structures with bounded stretch factors, bounded node degree, or planar structures. We also review the current status of controlling the transmission power so the total or maximum transmission power is minimized without sacrificing the network connectivity. After reviewing the geometric structures, we review the so-called localized routing methods in Section 6.3. Location service protocols are also discussed. In Section 6.4, we review the current status of applying stochastic geometry to study the connectivity, capacity, and so on of wireless networks. We conclude in Section 6.5 by pointing out some possible research questions.

6.2 NETWORK TOPOLOGY CONTROL

We consider a wireless ad hoc network consisting of a set V of n wireless nodes distributed in a two-dimensional plane. By a proper scaling, we assume that all nodes have the maximum transmission range equal to one unit. These wireless nodes define a *unit disk graph* UDG(V) in which there is an edge between two nodes if and only if their Euclidean distance is at most one. In this chapter, we concentrate on how to apply some structural properties of a point set for wireless networks as we treat wireless devices as two-dimensional points.

Due to the limited resources of the wireless nodes, it is preferred that the underlying network topology be constructed in a localized manner. Stojmenovic et al. first defined what a localized algorithm is in [16, 19]. Here, a distributed algorithm constructing a graph G is a *localized algorithm* if every node u can exactly decide all edges incident on u based only on the information of all nodes within a constant number of hops of u (plus a constant number of additional nodes' information if necessary).

Energy conservation is a critical issue in ad hoc wireless network for the node and network life, as the nodes are powered by batteries only. In the most common power-attenuation model, the power, denoted by $p(e)$, needed to support a link $e = uv$ is $\|uv\|^{\beta}$, where $\|uv\|$ is the Euclidean distance between u and v, and β is a real constant between 2 and 5, depending on the wireless transmission environment. This power consumption is typically called *path loss*. Practically, there is some other overhead cost for each device to receive and then process the signal. For simplicity, this overhead cost can be integrated into one cost c, which is almost the same for all nodes. Without specification, it is assumed that $c = 0$.

Let $G = (V, E)$ be a n-vertex connected weighted graph over V. The distance in G between two vertices $u, v \in V$ is the total weight of the shortest path between u and v and is denoted by $d_G(u, v)$. A subgraph $H = (V, E')$, where $E' \subseteq E$, is a *t-spanner* of G if for every $u, v \in V$, $d_H(u, v) \leq t \cdot d_G(u, v)$. The value of t is called the *stretch factor*. If the weight is the length of the link, then t is called the length stretch factor; if the weight is the power to support the communication of the link, then t is called the power stretch factor.

Recently, topology control for wireless ad hoc networks has attracted considerable attention [20–28]. Rajaraman [29] conducted an excellent survey recently.

6.2.1 Known Structures

Several geometrical structures have been studied recently both by computational geometry scientists and network engineers. Here we review the definitions of some of them that could be used in wireless networking applications.

Let V be the set of wireless nodes in a two dimensional plane. The *relative neighborhood graph,* denoted by RNG(V), is a geometric concept proposed by Toussaint [30]. It consists of all edges uv such that there is no point $w \in V$ with uw and wv satisfying $\|uw\| < \|uv\|$ and $\|wv\| < \|uv\|$. Let $disk(u, v)$ be the disk with diameter uv. Then, the *Gabriel graph* [31] GG(V) contains an edge uv from G if and only if $disk(u, v)$ contains no other vertex $w \in V$ inside. It is easy to show that RNG(V) is a subgraph of the Gabriel graph GG(V). For the unit disk graph, the relative neighborhood graph and the Gabriel graph only contain the edges in UDG satisfying the respective definitions.

The relative neighborhood graph has length stretch factor as large as $n - 1$ [32]. Li et al. [33] showed that its power stretch factor could also be as large as $n - 1$. The Gabriel graph has length stretch factor between $\sqrt{n}/2$ and $4\pi\sqrt{2n-4}/3$ [32]. Li et al. [33] showed that the power stretch factor of any Gabriel graph is exactly one when the overhead cost $c = 0$.

The *Yao graph* with an integer parameter $k \geq 6$, denoted by $\overrightarrow{YG_k}(V)$, is defined as follows. At each node u, any k equally separated rays originated at u define k cones. In each cone, choose the shortest edge uv, if there is any, and add a directed link $\overrightarrow{uv}$. Ties are broken arbitrarily or by the smallest ID. The resulting directed graph is called the Yao graph. See Figure 6.1 for an illustration. Let $YG_k(V)$ be the undirected graph by ignoring the direction of each link in $\overrightarrow{YG_k}(V)$. If we add the link $\overrightarrow{vu}$ instead of the link $\overrightarrow{uv}$, the graph is denoted by $\overleftarrow{YG_k}(V)$, which is called the *reverse* of the Yao graph. Some researchers used a similar construction called a θ-graph [34]. The difference is that, in each cone, it chooses the edge that has the shortest projection on the axis of the cone instead of the shortest edge. Here, the axis of a cone is the angular bisector of the cone. For more detail, please

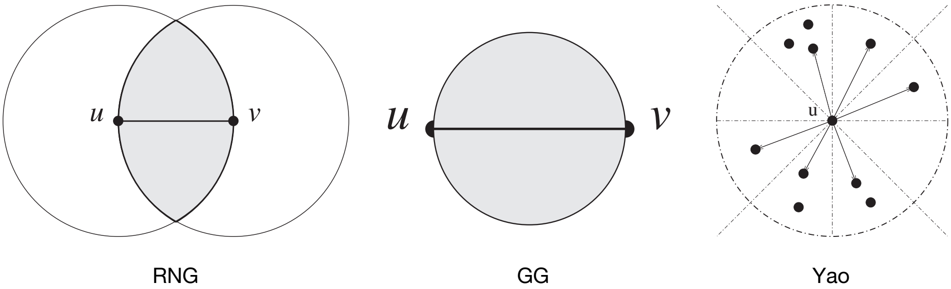


Figure 6.1. The definitions of RNG, GG, and Yao on point set. Left: The lune using uv is empty for RNG. Middle: The diametric circle using uv is empty for GG. Right: The shortest edge in each cone is added as a neighbor of u for Yao.

refer to [34]. Recently, the Yao structure has been rediscovered by several researchers for use in topology control in wireless ad hoc networks of directional antennas.

The Yao graph $YG_k(V)$ has length stretch factor $1/(1 - 2 \sin \pi/k)$. Li et al. [33] proved that the power stretch factor of the Yao graph $YG_k(V)$ is at most $1/[1 - (2 \sin \pi/k)^\beta]$.

Li et al. [35] extended the definitions of these structures on top of any given graph G. They proposed to apply the Yao structure on top of the Gabriel graph structure [the resulting graph is denoted by $\overrightarrow{YGG_k}(V)$], and apply the Gabriel graph structure on top of the Yao structure [the resulting graph is denoted by $\overrightarrow{GYG_k}(V)$]. These structures are sparser than the Yao structure and the Gabriel graph and they still have a constant bounded-power stretch factor. These two structures are connected graphs. Wattenhofer et al. [28] also proposed a two-phased approach that consists of a variation of the Yao graph followed by a variation of the Gabriel graph. Unfortunately, there are some bugs in their proofs, which were discussed in detail in [33].

Li et al. [23] proposed a structure that is similar to the Yao structure for topology control. Each node u finds a power $p_{u,\alpha}$ such that in every cone of degree α surrounding u, there is some node that u can reach with power $p_{u,\alpha}$. Here, nevertheless, we assume that there is a node reachable from u by the maximum power in that cone. Notice that the number of cones to be considered in the traditional Yao structure is a constant k. However, unlike the Yao structure, for each node u, the number of cones needed to be considered in the method proposed in [23] is about $2n$, where each node v could contribute two cones on both sides of segment uv. Then the graph G_α contains all edges uv such that u can communicate with v using power $p_{u,\alpha}$. They proved that, if $\alpha \leq 5\pi/6$ and the UDG is connected, then graph G_α is a connected graph. On the other hand, if $\alpha > 5\pi/6$, they showed that the connectivity of G_α is not guaranteed by giving some counterexample [23]. Unlike the Yao structure, the final topology G_α is not necessarily a bounded degree graph.

6.2.2 Bounded Node Degree

Notice that although the directed graphs $\overrightarrow{YG_k}(V)$, $\overrightarrow{GYG_k}(V)$, and $\overrightarrow{YGG_k}(V)$ have a bounded power-stretch factor and a bounded out-degree k for each node, some nodes may have a very large in-degree. The node configuration given in Figure 6.2 will result a very large in-degree for node u. The bounded out-degree gives us advantages when applied to several routing algorithms. However, an unbounded in-degree at node u will often cause large over-

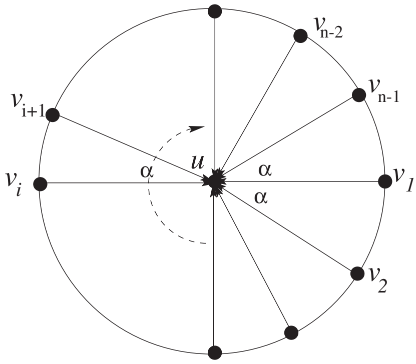


Figure 6.2. Node u has degree (or in-degree) $n - 1$.

head at u. 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 so as to be power-efficient.

Sink Structure. Arya et al. [36] gave an ingenious technique to generate a bounded degree graph with constant length-stretch factor. In [33], Li et al. applied the same technique to construct a sparse network topology with a bounded degree and a bounded power-stretch factor from $YG(V)$. The technique is to replace the directed star consisting of all links toward a node u by a directed tree $T(u)$ of a bounded degree with u as the sink. Tree $T(u)$ is constructed recursively. The algorithm is as follows.

Algorithm: Constructing-*T*(*u*) Tree[*u*, *I*(*u*)]

1. Choose k equal-sized cones $C_1(u), C_2(u), \cdots, C_k(u)$, centered at u.
2. Node u finds the nearest node $y_i \in I(u)$ in $C_i(u)$, for $1 \leq i \leq k$, if there is any. Link $\overrightarrow{y_iu}$ is added to $T(u)$ and y_i is removed from $I(u)$. For each cone $C_i(u)$, if $I(u) \cap C_i(u)$ is not empty, call Tree[y_i, $I(u) \cap C_i(u)$] and add the created edges to $T(u)$.

The union of all trees $T(u)$ is called the *sink structure* $\overrightarrow{YG}_k^*(V)$. Node u constructs the tree $T(u)$ and then broadcasts the structure of $T(u)$ to all nodes in $T(u)$. Since the total number of edges in the Yao structure is at most $k \cdot n$, where k is the number of cones divided, the total number of edges of $T(u)$ of all node u is also at most $k \cdot n$. Thus, the total communication cost is at most $k \cdot n$. Li et al. [33] proved that its power stretch factor is at most $\{1/[1 - (2 \sin \pi/k)^\beta]\}^2$, the maximum degree of the graph $\overrightarrow{YG}_k^*(V)$ is at most $(k + 1)^2 - 1$, and the maximum out-degree is k.

Notice that the sink structure and the Yao graph structure do not have to have the same number of cones, and the cones centered at different nodes do not need to be aligned. For setting up a power-efficient wireless network, each node u finds all its neighbors in $YG_k(V)$, which can be done in linear time proportional to the number of nodes within its transmission range.

YaoYao Structure. Li et al. [35] proposed another structure called YaoYao. Assume that each node v_i of V has a unique identification number $ID(v_i) = i$. The identity of a directed link $\overrightarrow{uv}$ is defined as $ID(\overrightarrow{uv}) = (\|uv\|, ID(u), ID(v))$.

Node u chooses a node v from each cone, if there is any, so the directed link $\overrightarrow{vu}$ has the smallest $ID(\overrightarrow{vu})$ among all directed links $\overrightarrow{wu}$ in $YG(V)$ in that cone. The union of all chosen directed links is the final network topology, denoted by $\overrightarrow{YY_k}(V)$. If the directions of all links are ignored, the graph is denoted as $YY_k(V)$. They [35] proved that the directed graph $\overrightarrow{YY_k}(V)$ is strongly connected if UDG(V) is connected and $k > 6$.

It was proved in [37] that $\overrightarrow{YY_k}(V)$ is a spanner in a civilized graph. Here a unit disk graph is a civilized graph if the distance between any two nodes in this graph is larger than a positive constant λ. In [38], the civilized unit disk graph is called the λ-precision unit disk graph. Notice that the wireless devices in wireless networks can not be too close or overlapped. Thus, it is reasonable to model the wireless ad hoc networks as a civilized unit disk graphs.

The experimental results obtained by Li et al. [35] showed that this sparse topology has a small power-stretch factor in practice. They [35] conjectured that $\overrightarrow{YY_k}(V)$ also has a constant bounded power-stretch factor theoretically in any unit disk graph. The proof of this conjecture or the construction of a counterexample remain to be determined.

Symmetric Yao Graph. In [35], Li et al. also considered another undirected structure called a *symmetric Yao graph* $YS_k(V)$. An edge uv is selected to graph $YS_k(V)$ if and only if both directed edges $\overrightarrow{uv}$ and $\overrightarrow{vu}$ are in the *Yao graph* $\overrightarrow{YG}_k(V)$. Then it is obvious that the maximum node degree is k.

Li et al. [33] showed that the graph $YS_k(V)$ is strongly connected if UDG(V) is connected and $k \geq 6$. The experiment by Li et al. also showed that $YS_k(V)$ has a small power-stretch factor in practice. However, it was shown recently in [21] that $YS_k(V)$ is not a spanner theoretically. The basic idea of the counterexample is similar to the counterexample for RNG proposed by Bose et al. [32].

High-Degree Yao Graph. Recently, Li et al. [39] proposed an efficient scatternet formation method based on the Yao structure.

The first step, which is optional, of the scatternet formation algorithm is to construct some subgraph satisfying some properties such as planar properties. In the second step, which is mandatory, of the algorithm, the degree of each node is limited to seven by applying the Yao structure, and the master–slave relations are formed in created subgraphs. Each node creates a key, which could be identity (ID), or degree, or the combination of both, for comparison with its neighbors.

In each iteration, undecided nodes with higher keys than any of their undecided neighbors (such nodes are referred as *active* nodes in the following) apply the Yao structure to limit the degree. They [39] described in detail how to assign master–slave relations. The active node then switches to a decided state. Assume that an active node u is a node that applies the Yao construction. Then node u divides the region surrounding it into seven equal angles centered at u, and chooses the closest node from each region, if there is any, with ties broken arbitrarily. All remaining connections at u are simply deleted from the graph. Notice that the elimination of any such edge uv by u immediately reduces the degree of v, that is, node v has to remove link uv also. However, in order to avoid excessive information exchange between neighbors, the originally decided keys (that is, original degrees) are used in all comparisons. We call the final structure as $YH_k(S)$.

This structure $YH_k(S)$ is different from all previous structures. First of all, $YS_k(S) \subseteq YH_k(S)$ since any edge uv from $YS_k(S)$ will not be removed by either node u or node v in the construction of $YH_k(S)$. It is not difficult to construct an example, for example, that in

Figure 6.3, such that $YS_k(S) \neq YH_k(S)$. The right two figures of Figure 6.3 also show that $YH_k(S)$ is different from $YY_k(S)$.

6.2.3 Planar Spanner

The Gabriel graph was used as a planar subgraph in the Face routing protocol [16, 40, 41] and the GPSR routing protocol [17]. The right-hand rule is used to guarantee the delivery of the packet in [16]. The relative neighborhood graph RNG was used for efficient broadcasting (minimizing the number of retransmissions) in the one-to-one broadcasting model in [42]. Since RNG and GG have large stretch factors in the worst case, some other structure is needed if we want to bound the distance traveled from the source to the destination. One of the known planar spanners is the Delaunay triangulation.

Assume that there are no four vertices of V that are cocircular. A triangulation of V is a *Delaunay triangulation,* denoted by $Del(V)$, if the circumcircle of each of its triangles does not contain any other vertices of V in its interior. A triangle is called the *Delaunay triangle* if its circumcircle is empty of vertices of V. The *Voronoi region*, denoted by $Vor(p)$, of a vertex $p \in V$, is a collection of two-dimensional points such that every point is closer to p than to any other vertex of V. The *Voronoi diagram* for V is the union of all Voronoi regions $Vor(p)$, where $p \in V$. The Delaunay triangulation $Del(V)$ is also the dual of the Voronoi diagram: two vertices p and q are connected in $Del(V)$ if and only if $Vor(p)$ and $Vor(q)$ share a common boundary. The shared boundary of two Voronoi regions $Vor(p)$ and $Vor(q)$ is on the perpendicular bisector line of segment pq. The boundary segment of a Voronoi region is called the *Voronoi edge*. The intersection point of two Voronoi edges is called the *Voronoi vertex*. The Voronoi vertex is the circumcenter of some Delaunay triangle.

Given a set of nodes V, it is well known that the Delaunay triangulation $Del(V)$ is a planar t-spanner of the completed graph $K(V)$ [43–45]. However, it is not appropriate to require the construction of the Delaunay triangulation in the wireless communication environment because of the possible massive communications it requires. Given a set of points V, let UDel(V) be the graph formed by edges of $Del(V)$ with length at most one unit, that is, $UDel(V) = Del(V) \cap UDG(V)$. Li et al. [46] considered the *unit Delaunay triangulation* UDel(V) for the planar spanner of UDG. Using the approach from [45], Li et al. [46] proved that UDel(V) is a t-spanner of the unit disk graph UDG(V).

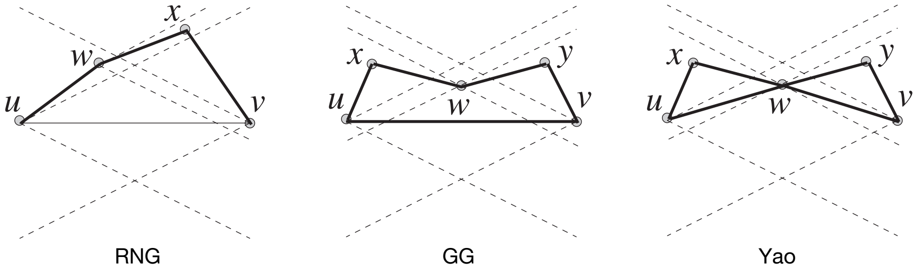


Figure 6.3. Left: The graph $YH_k(S)$ (represented by four solid lines) is different from $YS_k(S)$ (represented by three thick solid lines). Here the dashed lines define some cones around the nodes. Middle and Right: The graph $YH_k(S)$ is different from $YY_k(S)$. Here the node degrees are in decreasing order as w, u, v, x, and y.

6.2.3.1 Localized Delaunay Triangulation. Li et al. [46] gave a localized algorithm that constructs a sequence graphs, called a *localized Delaunay graph* $LDel^{(1)}(V)$, which are supergraphs of UDel(V). We begin with some necessary definitions before presenting the algorithm. Triangle Δuvw is called a *k-localized Delaunay triangle* if the interior of the circumcircle of Δuvw, denoted by $disk(u, v, w)$ hereafter, does not contain any vertex of V that is a k-neighbor of u, v, or w; and all edges of the triangle Δuvw have length no more than one unit. The *k-localized Delaunay graph* over a vertex set V, denoted by $LDel^{(k)}(V)$, has exactly all unit Gabriel edges and edges of all k-localized Delaunay triangles.

When it is clear from the context, we will omit the integer k in our notation of $LDel^{(k)}(V)$. As shown in [46], the graph $LDel^{(1)}(V)$ may contain some edges intersecting, although they showed that $LDel^{(1)}(V)$ can be decomposed to two planar graphs, that is, has thickness 2. They proved that $LDel^{(k)}(S)$ is a planar graph for any $k \geq 2$. Although the graph $UDel(V)$ is a t-spanner for $UDG(V)$, it is unknown how to construct it locally. Li et al. [46] present an efficient algorithm to extract a planar graph $PLDel(V)$ out of $LDel^{(1)}(V)$. They provide a novel algorithm to construct $LDel^{(1)}(V)$ using linear communications and then make it planar in linear communication cost. The final graph still contains $UDel(V)$ as a subgraph. Thus, it is a t-spanner of the unit-disk graph $UDG(V)$.

The basic approach of their method is to let each node u compute the Delaunay triangulation $Del[N_1(u)]$ of its 1-neighbors $N_1(u)$, including u itself. Node u then sends messages to its neighbors asking if the triangles in $Del[N_1(u)]$ can be accepted by $LDel^{(1)}(V)$. Its neighbor v accepts the triangle if it is in $Del[N_1(v)]$. The novel part is to bound the communications by only letting u query for triangle Δvuw if $\angle vuw$ is at least $\pi/3$. It was proved that the graph constructed by the above algorithm is $LDel^{(1)}(V)$. As $Del[N_1(u)]$ is a planar graph, and a proposal is made only if $\angle wuv \geq \pi/3$, node u broadcasts at most six proposals. And each proposal is replied to by at most two nodes. Therefore, the total communication cost is $O(n)$. They also gave an algorithm to extract from $LDel^{(1)}(V)$ a planar subgraph, denoted by $PLDel(V)$. They showed that $PLDel(V)$ is a supergraph of $LDel^{(2)}(V)$.

Recently, Cālinescu [47] proposed an efficient approach to collect $N_2(u)$ using total $O(n)$ communications based an efficient construction of the connected dominating set [7, 37]. Using the collected two-hop information, we then can construct the local Delaunay triangulation $LDel^{(2)}(V)$, which is guaranteed to be a planar graph. The cost of updating the structure $LDel^{(2)}(V)$ in a mobile environment could be more expensive than that of updating the structure $LDel^{(1)}(V)$ from the definition of these two structures. It remains open whether we can update these two structures using asymptotically same communication costs.

6.2.3.2 Restricted Delaunay Graph. Gao et al. [48] also proposed another structure, called a *restricted Delaunay graph* RDG, and showed that it has good spanning ratio properties and is easy to maintain locally. A restricted Delaunay graph of a set of points in the plane is a planar graph and contains all the Delaunay edges with length at most one. In other other words, they call any planar graph containing $UDel(V)$ a restricted Delaunay graph. They described a distributed algorithm to maintain the RDG such that at the end of the algorithm, each node u maintains a set of edges $E(u)$ incident to u. Those edges $E(u)$ satisfy that (1) each edge in $E(u)$ has length at most one unit; (2) the edges are consistent, that is, an edge $uv \in E(u)$ if and only if $uv \in E(v)$; (3) the graph obtained is planar; and (4) $UDel(V)$ is in the union of all edges $E(u)$.

The algorithm works as follows. First, each node u acquires the position of its one-hop neighbors $N_1(u)$ and computes the Delaunay triangulation $Del[N_1(u)]$ on $N_1(u)$, including u itself. In the second step, each node u sends $Del[N_1(u)]$ to all of its neighbors. Let $E(u) = \{uv \mid uv \in Del[N_1(u)]\}$. For each edge $uv \in E(u)$, and for each $w \in N_1(u)$, if u and v are in $N_1(w)$ and $uv \notin \mathrm{Del}[N_1(w)]$ then node u deletes edge uv from $E(u)$.

They proved that when the above steps are finished, the resulting edges $E(u)$ satisfy the four properties listed above. However, unlike the local Delaunay triangulation, the computation cost and communication cost of each node needed to obtain $E(u)$ is not optimal within a small constant factor. The communication cost could be as large as $\Theta(n^2)$, and the computation cost could be as large as $\Theta(n^3)$.

6.2.3.3 Partial Delaunay Triangulation.

Stojmenovic and Li [39] also proposed a geometry structure, namely the partial Delaunay triangulation (*PDT*), that can be constructed in a localized manner. Partial Delaunay triangulation contains a Gabriel graph as its subgraph, and itself is a subgraph of the Delaunay triangulation; more precisely, the subgraph of the unit Delaunay triangulation UDel(V). The algorithm for the construction of PDT goes as follows.

Let u and v be two neighboring nodes in the network. Edge uv belongs to $Del(V)$ if and only if there exists a disk with u and v on its boundary, which does not contain any other point from the set V. First test whether $disk(u, v)$ contains any other node from the network. If it does not, the edge belongs to *GG* and therefore to *PDT*. If it does, check whether nodes exist on both sides of line uv or on only one side. If both sides of line uv contain nodes from the set inside $disk(u, v)$ then uv does not belong to $Del(V)$.

Suppose now that only one side of line uv contains nodes inside the circle $disk(u, v)$, and let w be one such point that maximizes the angle $\angle uwv$. Let $\alpha = \angle uwv$. Consider now the largest angle $\angle uxv$ on the other side of the mentioned circle $disk(u, v)$, where x is a node from the set S. If $\angle uwv + \angle uxv > \pi$, then edge uv is definitely not in the Delaunay triangulation $Del(V)$. The search can be restricted to common neighbors of u and v, if only one-hop neighbor information is available, or to neighbors of only one of the nodes if two-hop information (or exchange of the information for the purpose of creating PDT is allowed) is available. Then whether edge uv is added to *PDT* is based on the following procedure.

Assume that only $N_1(u)$ is known to u, and there is one node w from $N_1(u)$ that is inside $disk(u, v)$ with the largest angle $\angle uwv$. Edge uv is added to *PDT* if the following conditions hold: (1) there is no node from $N_1(u)$ that lies on the different side of uv with w and inside the circumcircle passing through u, v, and w; (2) $\sin \alpha > d/R$, where R is the transmission radius of each wireless node, d is the diameter of the circumcircle $disk(u, v, w)$, and $\alpha = \angle uwv$ (here $\alpha \geq \pi/2$).

Assume only one-hop neighbors are known to u and v, and there is one node w from $N_1(u) \cup N_1(v)$ that is inside $disk(u, v)$ with the largest angle $\angle uwv$ (Figure 6.4). Edge uv is added to *PDT* if the following conditions hold: (1) there is no node from $N_1(u) \cup N_1(v)$ that lies on the different side of uv with w and inside the circumcircle passing u, v, and w; (2) $\cos \alpha/2 > d/2R$, where R is the transmission radius of each wireless node and $\alpha = \angle uwv$.

Obviously, the partial Delaunay triangulation is a subgraph of $UDel(V)$. The spanning ratio of the partial Delaunay triangulation could be very large.

Hu [22] proposed a structure using the Delaunay triangulation to bound the node degree of each wireless node to be at most Δ. The centralized algorithm starts with the Delaunay triangulation $Del(S)$ of the set of wireless nodes and then removes the edges in

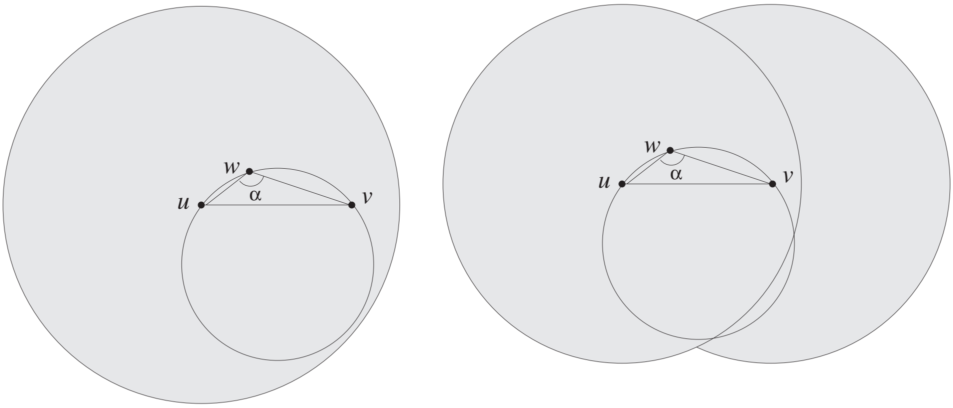


Figure 6.4. Left: Only one-hop information is known to u. Then $disk(u, v, w)$ must be covered by the transmission range of u (denoted by the shaded region) and is empty of neighbors of u. Right: Node u knows $N_1(u)$ and node v knows $N_1(v)$. The circumcircle $disk(u, v, w)$ is covered by the union of the transmission ranges of u and v and is empty of other vertices.

$Del(S)$ that are longer than the transmission range (normalized to one unit here). Then it processes the remaining edges in the order of the decreasing length and removes an edge if it causes either end node to have degree larger than Δ. Finally, it processes in the order of increasing length all possible edges that are not in the graph, and adds an edge if it does not cause a violation of the degree constraint. The worst time complexity of the above approach is $O(n^2 \log n)$. In [22], a distributed implementation was also proposed. Unfortunately, it is not correct since it requires each node u to find Delaunay edges uv with $\|uv\| \leq 1$. However, if we replace the computation of the Delaunay edges with length at most one unit by the computation of the local Delaunay triangulation, the method still produce a planar structure with bounded degree.

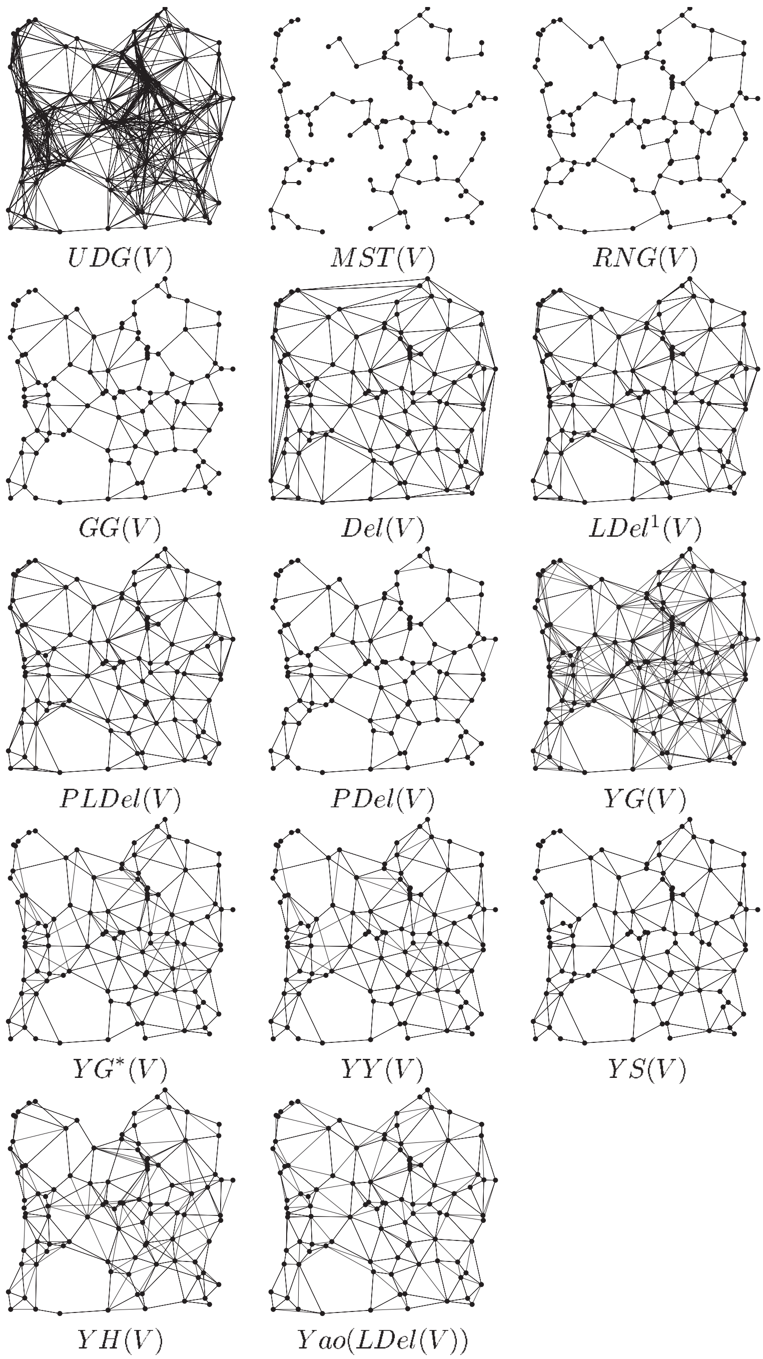
Recently, Li and Wang [49] proposed a novel localized method to construct a bounded-degree planar spanner for wireless ad hoc networks using $O(n)$ total communications.

Although all the structures discussed so far are flat structures, there are another set of structures, called hierarchical structures, that are used in wireless networks. Instead of all nodes being involved in relaying packets for other nodes, the hierarchical routing protocols pick a subset of nodes that serve as the routers, forwarding packets for other nodes. The structure used to build this virtual backbone is usually the connected dominating set. See a recent survey [9] on methods of constructing connected dominating sets efficiently in wireless ad hoc networks.

Figure 6.5 gives some concrete examples of the geometry structures introduced before. Here we randomly generate 100 nodes in a 200 m by 200 m square. The transmission radius of each node is set as 50 m. Notice that the graph LDel shown in Figure 6.5 is not a planar graph.

6.2.4 Transmission Power Control

In the previous sections, we have assumed that the transmission power of every node is equal and is normalized to one unit. We relax this assumption for a moment in this sub-



$UDG(V)$ $MST(V)$ $RNG(V)$

$GG(V)$ $Del(V)$ $LDel^1(V)$

$PLDel(V)$ $PDel(V)$ $YG(V)$

$YG^*(V)$ $YY(V)$ $YS(V)$

$YH(V)$ $Yao(LDel(V))$

Figure 6.5. Different topologies from *UDG*(*V*).

section. In other words, we assume that each node can adjust its transmission power according to its neighbors' positions for possible energy conservation. A natural question is then how to assign the transmission power for each node such that the wireless network is connected with optimization criteria being minimizing the maximum or total transmission power assigned.

A transmission power assignment on the vertices in V is a function f from V into real numbers. The *communication graph,* denoted by G_f, associated with a transmission power assignment f, is a directed graph with V as its vertices and has a directed edge $\overrightarrow{v_iv_j}$ if and only if $\|v_iv_j\|^\beta + c \le f(v_i)$. We call a transmission power assignment f *complete* if the communication graph G_f is strongly connected. Recall that a directed graph is strongly connected if, for any given pair of ordered nodes s and t, there is a directed path from s to t.

The *maximum cost* of a transmission power assignment f is defined as $mc(f) = \max_{v_i \in V} f(v_i)$. And the *total cost* of a transmission power assignment f is defined as $sc(f) = \Sigma_{v_i \in V} f(v_i)$. The min–max assignment problem is then to find a complete transmission power assignment f whose cost $mc(f)$ is the least among all complete assignments. The min–total assignment problem is to find a complete transmission power assignment f whose cost $sc(f)$ is the least among all complete assignments.

Given a graph H, we say the power assignment f is induced by H if

$$f(v) = \max_{(v,u)\in E} \|vu\|^\beta + c$$

where E is the set of edges of H. In other words, the power assigned to a node v is the largest power needed to reach all neighbors of v in H.

Transmission power control has been well studied by peer researchers in the recent years. Monks et al. [50] conducted simulations that show that implementing power control in a multiple access environment can improve the throughput of the nonpower-controlled IEEE 802.11 by a factor of 2. Therefore it provides a compelling reason for adopting the power controlled MAC protocol in wireless networks.

The min–max assignment problem was studied by several researchers [26, 51]. Let EMST(V) be the Euclidean minimum spanning tree over a point set V. Both [26] and [51] use the power assignment induced by EMST(V). It was proved in [26] that the longest edge of the Euclidean minimum spanning tree EMST(V) is always the critical link for min–max assignment. Here, for an optimum transmission power assignment f_{opt}, call a link uv the *critical link* if $\|uv\|^\beta + c = mc(f_{\text{opt}})$. Both algorithms presented in [26] and [51] compute the minimum spanning tree from the fully connected graph with possible very large communication cost. Notice that the best distributed algorithm [52–54] can compute the minimum spanning tree in $O(n)$ rounds using $O(m + n \log n)$ communications for a general graph with m edges and n nodes. Using the fact that the relative neighborhood graph, the Gabriel graph, and the Yao graph all have $O(n)$ edges and contain the Euclidean minimum spanning tree, a simple $O(n \log n)$ time-complexity centralized algorithm can be developed and can be implemented efficiently in a distributed manner.

The min–total assignment problem was studied by Kiroustis et al. [55] and by Clementi et al. [56–58]. Kiroustis et al. [55] first proved that the min–total assignment problem is *NP-hard* when the mobile nodes are deployed in a three-dimensional space. A simple two-approximation algorithm based on the Euclidean minimum spanning tree was also given in [55]. The algorithm guarantees the same approximation ratio in any dimensions.

Clementi et al. [56–58] proved that the min–total assignment problem is still NP-hard when nodes are deployed in a two-dimensional space.

So far, we have generated asymmetric communication graphs from the power assignment. For the symmetric communication, several methods also guarantee a good performance. It is easy to show that the minimum spanning tree method still gives the optimum solution for the min–max assignment and a two-approximation for the min–total assignment. Recently, Călinescu et al. [59] gave a method that achieves a better approximation ratio (5/3) by using ian dea from the minimum Steiner tree. Like the minimum spanning tree method, it works for any power definition.

6.3 LOCALIZED ROUTINGS

The geometric nature of the multihop ad hoc wireless networks allows a promising idea: localized routing protocols. A routing protocol is *localized* if the decision to which node to forward a packet is based only on:

- The information in the header of the packet. This information includes the source and the destination of the packet, but more data could be included, provided that its total length is bounded.
- The local information gathered by the node from a small neighborhood. This information includes the set of one-hop neighbors of the node, but a larger neighborhood set could be used provided it could be collected efficiently.

Randomization is also used in designing the protocols. A routing is said to be *memoryless* if the decision as to which node to forward a packet is solely based on the destination, current node, and its neighbors within some constant hops. Localized routing is sometimes called in the literature *stateless* [17], *online* [60, 61], or *distributed* [62].

6.3.1 Location Service

In order to make the localized routing work, the source node has to learn the current (or approximately current) location of the destination node. Notice that for sensor networks collecting data, the destination node is often fixed; thus, location service is not needed in these applications. However, the help of a *location service* is needed in most application scenarios. Mobile nodes register their locations to the location service. When a source node does not know the position of the destination node, it queries the location service to get that information. In cellular networks, there are dedicated position severs. It will be difficult to implement the centralized approach of location services in wireless ad-hoc networks. First, for centralized approach, each node has to know the position of the node that provides the location services, which is a chicken-and-egg problem. Second, the dynamic nature of the wireless ad hoc networks makes it very unlikely that there is at least one location server available for each node. Thus, we will concentrate on distributed location services.

For the wireless ad hoc networks, the location service provided can be classified into four categorizes: *some-for-all, some-for-some, all-for-some,* and *all-for-all.* Some-for-all service means that some wireless nodes provide location services for all wireless nodes. Other categorizations are defined similarly.

An example of all-for-all services is the location services provided in the Distance Routing Effect Algorithm for Mobility (DREAM) by Basagni et al. [63]. Each node stores a database of the position information for all other nodes in the wireless networks. Each node will regularly flood packets containing its position to all other nodes. A frequency of the flooding and the range of the flooding is used as a control of the cost of updating and the accuracy of the database.

Using the idea of *quorum* developed in the databases and distributed systems, Hass and Liang [64] and Stojmenovic [65] developed quorum-based location services for wireless ad hoc networks. Given a set of wireless nodes V, a quorum system is a set of subset (Q_1, $Q_2, \cdots, Q_k$) of nodes whose union is V. These subsets could be mutually disjoint or often have equal numbers of intersections. When one of the nodes requires the information of the other, it suffices to query one node (called the representative node of Q_i) from each quorum Q_i. A virtual backbone is often constructed between the representative nodes using a nonposition-based methods such as those in [7, 6]. The updating information of a node v is sent to the representative node (or the nearest if there are many) of the quorum containing v. The difficulty of using quorums is that the mobility of the nodes requires the frequent updating of the quorums. The quorum-based location service is often of the *some-for-some* type.

The other promising location service is based on the quadtree partition of the two-dimensional space [66]. It divides the region containing the wireless network into a hierarchy of squares. The partition of the space in [66] is uniform. However, we notice that the partition could be nonuniform if the density of the wireless nodes were not uniform for some applications. Each node v will have the position information of all nodes within the same *smallest* square containing v. This position information of v is also propagated to up-layer squares by storing it in the node with the nearest identity to v in each up-layer square containing v. Using the nearest identity over the smallest identity can avoid the overload of some nodes. The query is conducted accordingly. It is easy to show that it takes about $O(\log n)$ time to update the location of v and to query another node's position information.

6.3.2 Localized Routing Protocols

We summarize some localized routing protocols proposed in the networking and computational geometry literature (see Figure 6.6).

The following routing algorithms on the graphs were proposed recently.

Compass Routing. Let t be the destination node. Current node u finds the next relay node v such that the angle $\angle vut$ is the smallest among all neighbors of u in a given topology. See [67].

Random Compass Routing. Let u be the current node and t be the destination node. Let v_1 be the node on the above of line ut such that $\angle v_1ut$ is the smallest among all such neighbors of u. Similarly, we define v_2 to be nodes below line ut that minimizes the angle $\angle v_2ut$. Then node u randomly chooses v_1 or v_2 to forward the packet. See [67].

Greedy Routing. Let t be the destination node. Current node u finds the next relay node v such that the distance $||vt||$ is the smallest among all neighbors of u in a given topology. See [16].

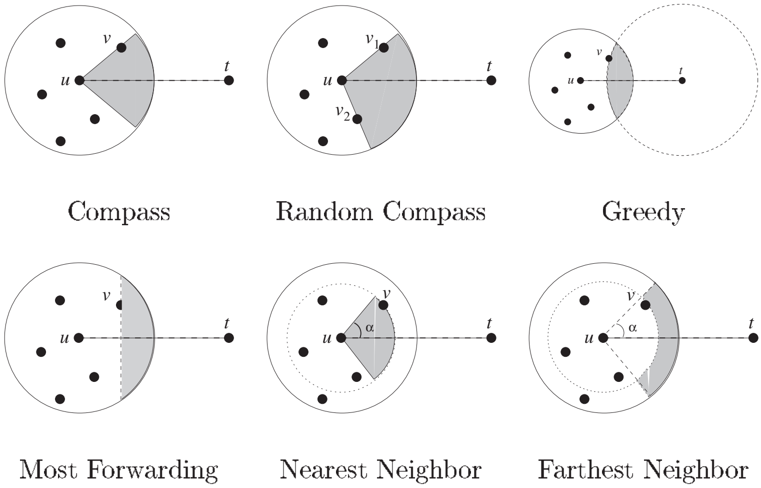


Figure 6.6. Various localized routing methods. Shaded area is empty and v is next node.

Most Forwarding Routing (MFR). Current node u finds the next relay node v such that $\|v't\|$ is the smallest among all neighbors of u in a given topology, where v' is the projection of v on segment ut. See [62].

Nearest Neighbor Routing (NN). Given a parameter angle α, node u finds the nearest node v as forwarding node among all neighbors of u in a given topology such that $\angle vut \leq \alpha$.

Farthest Neighbor Routing (FN). Given a parameter angle α, node u finds the farthest node v as forwarding node among all neighbors of u in a given topology such that $\angle vut \leq \alpha$.

Greedy–Compass. Current node u first finds the neighbors v_1 and v_2 such that v_1 forms the smallest counterclockwise angle $\angle tuv_1$ and v_2 forms the smallest clockwise angle $\angle tuv_2$ among all neighbors of u with the segment ut. The packet is forwarded to the node of $\{v_1, v_2\}$ with minimum distance to t. See [61, 68]

Notice that it is shown in [16, 67] that compass routing, random compass routing, and greedy routing guarantee to deliver the packets from the source to the destination if Delaunay triangulation is used as the network topology. They proved this by showing that the distance from the selected forwarding node v to the destination node t is less than the distance from current node u to t. However, the same proof cannot be carried over when the network topology is Yao graph, Gabriel graph, relative neighborhood graph, and localized Delaunay triangulation. When the underlying network topology is a planar graph, the right-hand rule is often used to guarantee the packet delivery after simple localized routing heuristics fail [16, 62, 17].

It was proved in [68] that greedy routing guarantees the delivery of the packets if the Delaunay triangulation is used as the underlying structure. Compass routing guarantees the delivery of the packets if the regular triangulation is used as the underlying structure. There are triangulations (not Delaunay) that defeat these two schemes. Greedy–compass routing guarantees the delivery of the packets as long as there is a triangulation used as the underlying structure. Every oblivious routing method is defeated by some convex subdivisions.

Localized routing protocols support mobility by eliminating the communication-intensive task of updating the routing tables. But mobility can affect the localized routing protocols, in both the performance and the guarantee of delivery. There has been no work so far to design protocols with guaranteed delivery when the network topology changes during the routing.

6.3.3 Quality Guaranteed Protocols

With respect to localized routing, there are several ways to measure the quality of the protocol. Given the scarcity of the power resources in wireless networks, minimizing the total power used is imperative. A stronger condition is to minimize the total Euclidean distance traversed by the packet. Morin et al. [61, 68] also studied the performance ratio of previously studied localized routing methods. They proved that none of the previous proposed heuristics guarantees a constant ratio of the traveled distance of a packet compared with the minimum. They gave the first localized routing algorithm such that the traveled distance of a packet from u to v is at most a constant factor of $\|uv\|$ when the Delaunay triangulation is used as the underlying structure.

Bose and Morin [68] basically use the binary search method to find which path of the tunnel connecting the source u and the destination v is better. However, their algorithm (called DTR hereafter) needs the Delaunay triangulation, which is expensive to construct in wireless ad hoc networks, as the underlying structure. In [69], they further extend their method to any triangulations satisfying the diamond property. Let $G(V, r_n)$ be the graph defined over V, which has an edge uv iff $\|uv\| \leq r_n$. Li et al. [70] showed that the sum of all edges in $Del(V)$ is no more than r_n with high probability, where r_n is the transmission radius needed by each node so the induced unit disk graph $G(V, r_n)$ is connected with high probability.

To make $G(V, r_n)$ connect with probability $1 - (1/n)$, we need $n\pi r_n^2 \geq 2 \ln n$, see [71]. They [70] showed that, with probability at least $1 - (1/\beta)$, the longest edge D_n of Delaunay triangulation is $D_n \leq \sqrt{3(\ln n + \ln \beta + \ln 3)/(n\pi)}$. Thus, the required transmission range so that local Delaunay triangulation $PLDel$ equals the Delaunay triangulation Del is just $\sqrt{3/2}$ of the minimum transmission range to have a connected network with high probability. This implies that the localized Delaunay triangulation can be used to approximate the Delaunay triangulation almost always when the network $G(V, r_n)$ is connected and when V is randomly deployed. Consequently, the method by Bose et al. [68] can be used on local Delaunay triangulation almost always.

Table 6.1 illustrates the delivery rates. For routing methods NN and FN, we choose the next node within $\pi/3$ of the destination direction. Interestingly, we found that when the Yao graph is used, the delivery rates are high in all methods. The reason these methods delivered the packets when the Yao structure is used could be that there is a node within the transmission range in the direction of the destination with high probability when $N_1(u)$ is large enough. Table 6.2 illustrates the maximum spanning ratios of the path traversed by the packet from source s to destination t to $\|st\|$. Although the maximum spanning ratio by DTR is larger than most previous methods, DTR is the only known method guaranteeing a constant spanning ratio.

Table 6.1. The Delivery Rate

	Yao	RNG	GG	Del	LDel[(2)]	PLDel
NN	100	20.4	83.3	100	100	98.4
FN	94.5	25.8	70.2	94.6	100	95.2
MFR	98.6	54.5	90	97.4	95	97.2
Cmp	97.1	23.2	66.2	100	100	100
RCmp	95.4	51.4	66.2	85.7	86.9	89.9
Grdy	100	78.3	100	100	100	100
GCmp	95	26.5	76.6	100	98.4	100
DTR				100	100	100

Table 6.2. The Maximum Spanning Ratio

	Yao	RNG	GG	Del	LDel[(2)]	PLDel
NN	1.7	1.3	1.6	1.5	1.6	1.6
FN	3.3	1.6	1.8	1.9	2.3	2.4
MFR	4.7	1.7	2.3	1.8	1.8	3.1
Cmp	11	2.2	6.2	16	18	18
RCmp	27	20	19	31	27	21
Grdy	1.7	2.0	1.8	1.6	1.5	1.6
GCmp	2.5	1.6	2.7	1.9	1.9	1.9
DTR				8.6	8.6	8.4

6.4 STOCHASTIC GEOMETRY

One of the remaining fundamental and critical issues is to have multiple disjoint paths connecting every pair of nodes without sacrificing the spectrum-reusing property. As power is a scarce resource in wireless networks, it is important to save the power consumption without losing the network connectivity. The universal minimum power used by all wireless nodes such that the induced network topology is connected is called the *critical power*. Although determining the critical power for static wireless ad hoc networks has been well studied [26, 51, 72], it remains to study the critical power for connectivity for mobile wireless networks. As the wireless nodes move around, it is impossible to have a unanimous critical power to guarantee the connectivity for all instances of the network configuration. Thus, we need to find a critical power, if possible, at which each node has to transmit to guarantee the connectivity of the network almost surely, i.e., with high probability of almost one.

For simplicity, we assume that the wireless devices are distributed in a unit-area square (or disk) according to some distribution function, e.g., uniform distribution or Poisson process. A point set process is said to be a *random uniform point process,* denoted by $\mathcal{X}_n$, in a unit-area square $\mathcal{C} = [-0.5, 0.5] \times [-0.5, 0.5]$ if it consists of n independent points each of which is uniformly and randomly distributed over $\mathcal{C}$.

The standard probabilistic model of a *homogeneous Poisson process* with density n, denoted by $\mathcal{P}_n$, is characterized by the property that the number of nodes in a region is a random variable depending only on the area (or volume in higher dimensions) of the region. In other words,

- The probability that there are exactly k nodes appearing in any region Ψ of area A is $[(n/A)^k/k] \cdot e^{-nA}$.
- For any region Ψ, the conditional distribution of nodes in Ψ given that exactly k nodes in the region is *joint uniform*.

Hereafter, we assume that the movement of wireless devices still keeps them in the same distribution (uniform or Poisson process). Gupta and Kumar [72] showed that there is almost surely a critical power at which the wireless nodes are randomly and uniformly distributed in a unit area disk. The result by Penrose [71] implies the same conclusion. Moreover, Penrose [71] gave the probability of the network to be connected if the transmission radius is set as a positive real number r and n goes to infinity.

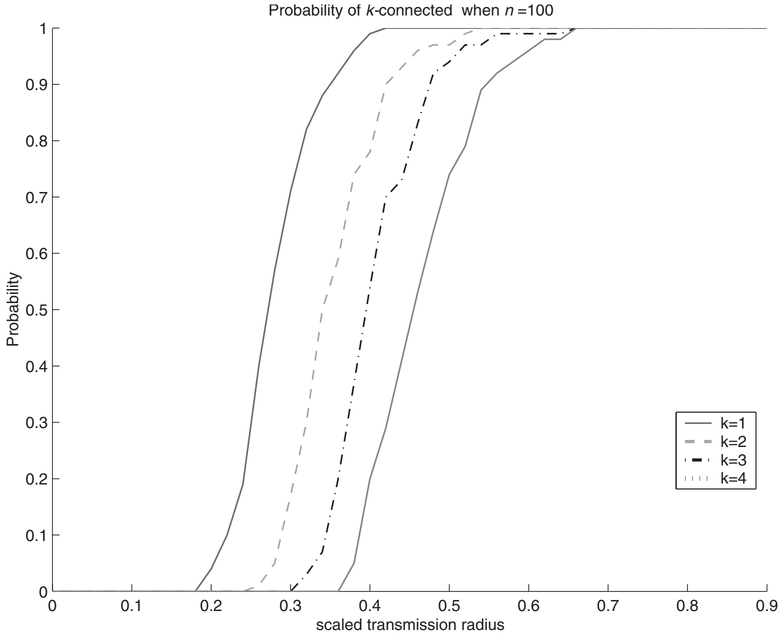
The theoretical value of the transmission ranges gives us insight into how to set the transmission radius to achieve the k-connectivity with a certain probability. These results also apply to mobile networks in which the moving of wireless nodes always generates randomly (or Poisson process) distributed node positions. They also have applications in system design of large scale wireless networks. For example, for setting up a sensor network monitoring a certain region, we should deploy how many sensors in order to have a multiple connected network, knowing that each sensor can transmit a range r_0? Notice that most results hold only when the number of wireless devices n goes to infinity, which is difficult to deploy practically. Li et al. [73] conducted extensive simulations to study the transmission radius achieving k-connectivity with certain probability for practical settings.

Let $G(V, r)$ be the graph defined on V with edges $uv \in E$ if and only if $\|uv\| \leq r$. Here $\|uv\|$ is the Euclidean distance between nodes u and v. Let $\mathcal{G}(\mathcal{X}_n, r_n)$ be the set of graphs $G(V, r_n)$ for n nodes V that are uniformly and independently distributed in a two-dimensional unit-area disk $\mathcal{D}$ with center at the origin. The problem considered by Gupta and Kumar [72] is then to determine the value of r_n such that a random graph in $\mathcal{G}(\mathcal{X}_n, r_n)$ is asymptotically connected with probability one as n goes to infinity. Let $P_k(\mathcal{X}_n, r_n)$ be the probability that a graph in $\mathcal{G}(\mathcal{X}_n, r_n)$ is k-connected.

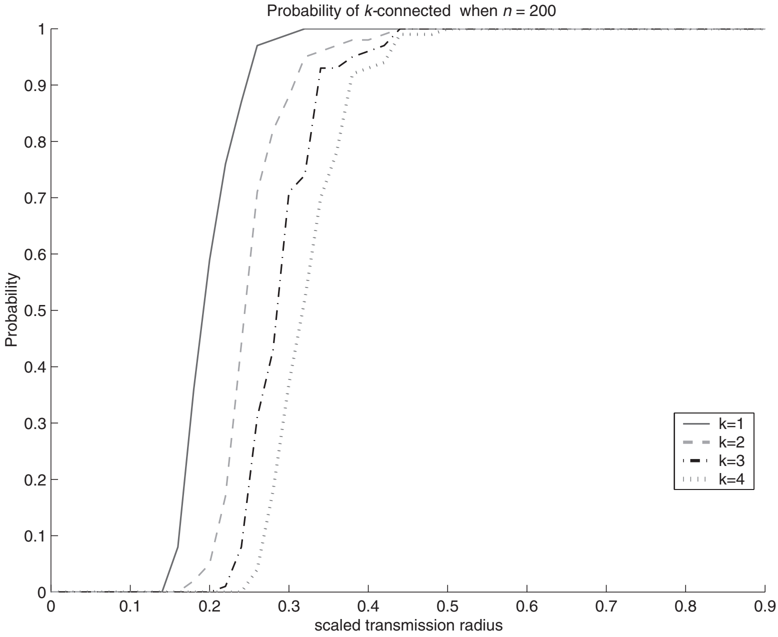
Fault tolerance is one of the central challenges in designing wireless ad hoc networks. To make fault tolerance possible, first of all, the underlying network topology must have multiple disjoint paths to connect any two given wireless devices. Here, the path could be vertex disjoint or edge disjoint. Considering the communication nature of the wireless networks, the vertex disjoint multiple paths are often used in the literature. Here, we are interested in what is the condition of r_n such that the underlying network topology $G(V, r_n)$ is k-connected almost surely when V is uniformly and randomly distributed over a two-dimensional domain Ω. A graph is called k-vertex connected (k-connected for simplicity) if, for each pair of vertices, there are k mutually vertex disjoint paths (except end-vertices) connecting them. Equivalently, a graph is k-connected if there is no set of $k - 1$ nodes whose removal will partition the network into at least two components. Thus, a k-connected wireless network can sustain the failure of $k - 1$ nodes.

The *vertex connectivity,* denoted by $\kappa(G)$, of a graph G is the maximum k such that G is k-vertex connected. The *edge connectivity,* denoted by $\xi(G)$, of a graph G is the maximum k such that G is k-edge connected. The minimum degree of a graph G is denoted by $\delta(G)$ and the maximum degree of a graph G is denoted by $\Delta(G)$. Clearly, for any graph G, $\kappa(G) \leq \xi(G) \leq \delta(G) \leq \Delta(G)$.

A graph property is called *monotone increasing* if G has such a property and all graphs on the same vertex set containing G as a subgraph have this property. Let $\mathcal{Q}$ be any monot-

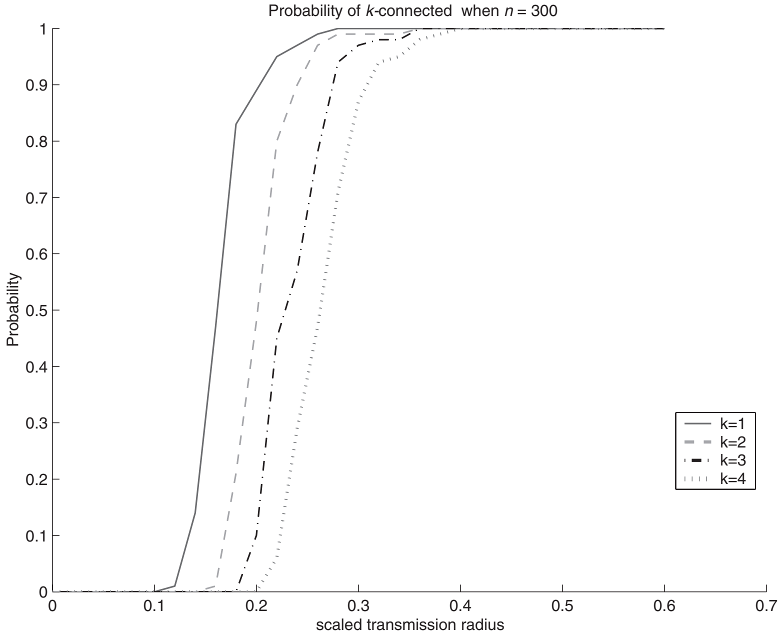


$n = 100$

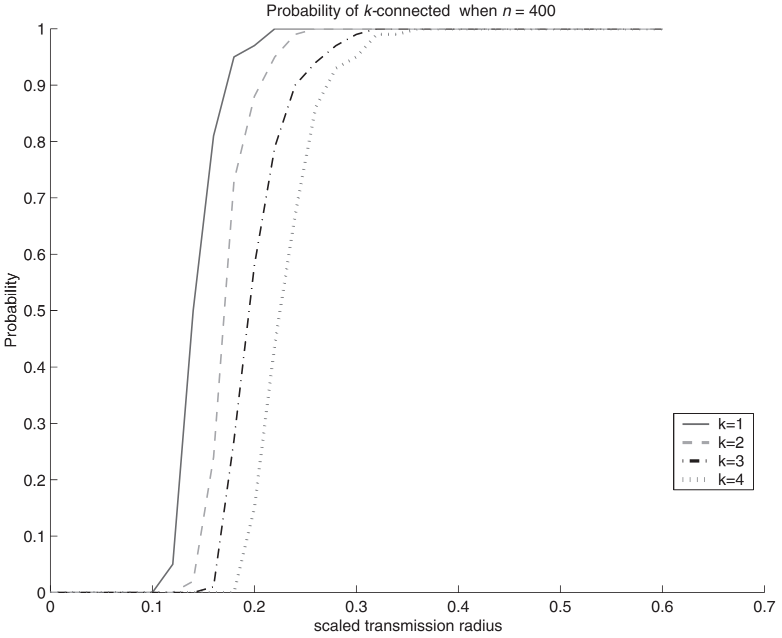


$n = 200$

Figure 6.7. Transition phenomena of graph $G(V, r)$ being k-connected.



n = 300



n = 400

Figure 6.7. *Continued.*

one increasing property of graphs, for example, the connectivity, the k-edge connectivity, the k-vertex connectivity, the minimum node degree at least k, and so on. The *hitting radius* $\varrho(V, Q)$ is the infimum of all r such that graph $G(V, r)$ has property Q. For example, $\varrho(V, \kappa \geq k)$ is the minimum radius r such that $G(V, r)$ is at least k-vertex connected; $\varrho(V, \delta \geq k)$ is the minimum radius r at which the graph $G(V, r)$ has the minimum degree k. Obviously, for any V,

$$\varrho(V, \kappa \geq k) \geq \varrho(V, \delta \geq k)$$

Penrose [74] showed that these two hitting radii are asymptotically the same for n points V randomly and uniformly distributed in a unit square and n goes infinity.

The connectivity of random graphs, especially the geometric graphs and their variations, have been considered in the random graph theory literature [75], in the stochastic geometry literature [71, 74, 76–78], and the wireless ad hoc network literature [72, 79–86].

Let us first consider the connectivity problem. Given n nodes V randomly and independently distributed in a unit-area disk $\mathcal{D}$, Gupta and Kumar [72] showed that $G(V, r_n)$ is connected almost surely if $n\pi \cdot r_n^2 \geq \ln n + c(n)$ for any $c(n)$ with $c(n) \to \infty$ as n goes infinity. Notice that this bound is tight, as they also proved that $\mathcal{G}(\mathcal{X}_n, r_n)$ is asymptotically disconnected with positive probability if $n\pi \cdot r_n^2 = \ln n + c(n)$ and $\lim\sup_n c(n) < +\infty$. Notice that they actually derived their results for a homogeneous Poisson process of points in $\mathcal{D}$ instead of the independent and uniform point process. They showed that the difference between them is negligible. Penrose [71] showed that the same result holds if the geometry domain in which the wireless nodes are distributed is a unit-area square $\mathcal{C}$ instead of the unit-area disk $\mathcal{D}$.

Independently, Penrose [71] showed that the longest edge M_n of the minimum spanning tree of n points randomly and uniformly distributed in a unit area square $\mathcal{C}$ satisfies

$$\lim_{n \to \infty} Pr(n\pi M_n^2 - \ln n \leq \alpha) = e^{-e^{-\alpha}}$$

for any fixed real number α. Here $Pr(X)$ is the probability of event X. Remember that the longest edge of EMST is always the critical power [26, 51]. Thus, the result in [71] is actually stronger than that in [72] since it will give the probability that the network is connected. For example, if we set $\alpha = \ln \ln n$, we have $Pr(n\pi M_n^2 \leq \ln n + \ln \ln n) = e^{-1/\ln n}$. It implies that the network is connected with probability at least $e^{-1/\ln n}$ if the transmission radius of each node r_n satisfies $n\pi r_n^2 = \ln n + \ln \ln n$. Notice that $e^{-1/\ln n} > 1 - (1 - \ln n)$ from $e^{-x} > 1 - x$ for $x > 0$. By setting $\alpha = \ln n$, the probability that the graph $G(V, r_n)$ is connected is at least $e^{-1/n} > 1 - (1/n)$, where $n\pi r_n^2 = 2 \ln n$. Notice that the above probability is only true when n goes to infinity. When n is a finite number, then the probability of the graph being connected is smaller. In [73], Li et al. presented the experimental study of the probability of the graph $G(V, r_n)$ being connected for finite number n.

One closely related question is the *coverage* problem: disks of radius r are placed in a two-dimensional unit-area disk $\mathcal{D}$ with centers from a Poisson point process with intensity n. A result shown by Hall [87] implies that, if $n\pi \cdot r^2 = \ln n + \ln \ln n + c(n)$ and $c(n) \to \infty$, then the probability that there is a vacancy area in $\mathcal{D}$ is 0 as n goes infinity; if $c(n) \to -\infty$, the probability that there is a vacancy in $\mathcal{D}$ is at least 1/20. This implies that the hitting radius r_n such that $G(V, r_n)$ is connected satisfies $\pi \cdot r_n^2 \leq 4[\ln n + \ln \ln n + c(n)/n]$ for $c(n) \to +\infty$.

Let $\mathcal{B}[n, p(n)]$ be the set of graphs on n nodes in which each edge of the completed graph K_n is chosen independently with probability $p(n)$. Then it has been shown that the probability that a graph in $\mathcal{B}[n, p(n)]$ is connected goes to one if $p(n) = [\ln n + c(n)/n]$ for any $c(n) \to \infty$. Although their asymptotic expressions are the same as those by Gupta and Kumar [72], we cannot apply this to the wireless model as, in wireless networks, the existences of two edges are not independent, and we do not choose edges from the completed graph using the Bernoulli model.

For geometric graphs, it was proved by Penrose [74] that, given any metric l_p with $2 \leq p \leq \infty$ and any positive integer k,

$$\lim_{n\to\infty} Pr[\varrho(\mathcal{X}_n, \kappa \geq k) = \varrho(\mathcal{X}_n, \delta \geq k)] = 1$$

The result is analogous to the well-known results in the graph theory [75] that a graph becomes k-vertex connected when it achieves the minimum degree k if we add the edges randomly and uniformly from $\binom{n}{2}!$ possibilities.

This result by Penrose [74] says that a graph of $G(\mathcal{X}_n, r)$ becomes k-connected almost surely at the moment it has minimum degree k by letting r go from 0 to ∞. However, this result does not imply that, to guarantee a graph over n points k-connected almost surely, we only have to connect every node to its k-nearest neighbors. Let V be n points randomly and uniformly distributed in a unit square (or disk). Xue and Kumar [86] proved that, to guarantee a geometry graph over V connected, the number of nearest neighbors that every node has to connect is asymptotically $\Theta(\ln n)$. Dette and Henze [76] studied the maximum length of the graph by connecting every node to its k nearest neighbors asymptotically. Li et al. [73] showed that, given n random points V over a unit-area square, to guarantee a geometry graph over V k-connected, the number of nearest neighbors that every node has to connect is asymptotically $\ln n + (2k - 3)\ln \ln n$.

Similarly, instead of considering $\mathcal{X}_n$, Penrose also considered a homogeneous Poisson point process with intensity n on the unit-area square $\mathcal{C}$. Penrose gave loose upper and lower bounds on the hitting radius $r_{n,k} = \varrho(\mathcal{P}_n, \delta \geq k)$ as $(\ln n/2^{d+1}) \leq nr^d_{n,k} \leq d!2 \ln n$ for a homogeneous Poisson point process on a d-dimensional unit cube, This result is too loose. More importantly, the parameter k does not appear in this estimation at all. In [73], a tighter bound on $r_{n,k}$ was derived for two-dimensional n points V randomly and uniformly distributed in $\mathcal{C}$ such that the graph $G(V, r_{n,k})$ is k-connected with high probability.

Bettstetter [79] conducted the experiments to study the relations of the k-connectivity and the minimum node degree using a toroidal model. Li et al. [73] also conducted experiments to study the probability that a graph has minimum degree k and has vertex connectivity k simultaneously using a Euclidean model. Surprisingly, they found that this probability is sufficiently close to 1 even when n is at the scale of 100. This observation implies a simple method (by just computing the minimum vertex degree) to approximate the connectivity of a random geometry graph. Recently, Bahramgiri et al. [20] showed how to decide the minimum transmission range of each node such that the resulting directed communication graph is k-connected. Here, it is assumed that the unit disk graph, by setting each node with the maximum transmission range, is k-connected. Lukovszki [88] provided a method to construct a spanner that can sustain k-nodes or k-links failures.

Penrose [71, 74] also studied the k-connectivity problem for d-dimensional points distributed in a unit-area cube using the toroidal model instead of the Euclidean model as one way to eliminate the boundary effects. He showed that the hitting radius $r_{n,k}$ such that the graph $G(V, r_{n,k})$ is k-connected satisfies

$$\lim_{n\to\infty} Pr[n\pi r_{n,k}^2 \le \ln n + (k-1)\ln\ln n - \ln(k-1)! + \alpha] = e^{-e^{-\alpha}}$$

Dette and Henze [76] studied the largest length, denoted by $\ell_{n,k}$ here, of the kth nearest neighbor link for n points drawn independently and uniformly from the d-dimensional unit-length cube or the d-dimensional unit volume sphere. They gave asymptotic results of this length according to $k < d$, $k = d$, or $k < d$. For unit-volume cube, they use the norm l_∞ instead of l_2. For the unit-volume sphere, their result implies that when $d = 2$ and $k > 2$,

$$\lim_{n\to\infty} Pr[n\pi\ell_{n,k}^2 \le \ln n + (2k-3)\ln\ln n - 2\ln(k-1)! - 2(k-2)\ln 2 + \ln\pi + 2\alpha] = e^{-e^{-\alpha}}$$

Notice that, Penrose [74] had showed that when the domain is a unit-area square, the probability that a random geometry graph $G(V, r_{n,k})$ is k-connected and has minimum vertex degree k goes to 1 as n goes to infinity. Following from a combination of [76] and [74], Li et al. [73] showed that if the transmission range $r_{n,k}$ satisfies $n\pi \cdot r_{n,k}^2 \ge \ln n + (2k-1)\ln \ln n - 2\ln k! + \alpha + 2\ln(8k/2^k\sqrt{\pi})$, then $G(V, r_{n,k})$ is $(k+1)$-connected with probability at least $e^{-e^{-\alpha}}$ as n goes to infinity.

6.5 CONCLUSION

Wireless ad hoc networks have attracted considerable attention recently due to their potential wide applications in various areas and, moreover, the ubiquitous computing. In this chapter, we presented an overview of the recent progress in topology control and localized routing in wireless ad hoc networks. Nevertheless, there are still many excellent results that are not covered in this survey due to space limitations.

There are many interesting open questions for topology control in wireless ad hoc networks. First, we would like to know whether the YaoYao structure $YY_k(V$ [and, similarly, the structure $YH_k(V)$] is a length spanner. Second, when the overhead cost c of signal transmission is not negligible, are the structures reviewed here still power spanners? Third, how can one control the network topology when different nodes have different transmission ranges such that the topology has some nice properties? Fourth, can we design a localized routing protocol that achieves constant ratio of the length of the found path to the minimum? The answer is probably negative; see [89].

ACKNOWLEDGMENT

The author would like to thank Dr. Ivan Stojmenovic for some helpful suggestions in preparing this chapter, and Dr. Mathew Penrose for reviewing it.

REFERENCES

1. S. Capkun, M. Hamdi, and J. P. Hubaux, “Gps-Free Positioning in Mobile Ad-Hoc Networks,” in *Proceedings of the Hawaii International Conference on System Sciences,* 2001.

2. P. Sinha, R. Sivakumar, and V. Bharghavan, “Cedar: Core Extraction Distributed Ad Hoc Routing,” in *Proceedings of IEEE INFOCOMM ’99,* 1999.
3. B. Das and V. Bharghavan, “Routing in Ad-Hoc Networks Using Minimum Connected Dominating Sets,” in *1997 IEEE International Conference on Communications (ICC’97),* 1997, vol. 1, pp. 376–380.
4. J. Wu and H. Li, “A Dominating-Set-Based Routing Scheme in Ad Hoc Wireless Networks,” *Telecommunication Systems Journal, 3,* 63–84, 2001.
5. I. Stojmenovic, M. Seddigh, and J. Zunic, “Dominating Sets and Neighbor Elimination Based Broadcasting Algorithms in Wireless Networks,” *IEEE Transactions on Parallel and Distributed Systems, 13,* 1, 14–25, 2002.
6. K. M. Alzoubi, P.-J. Wan, and O. Frieder, “New Distributed Algorithm for Connected Dominating Set in Wireless Ad Doc Networks,” in *Proceedings of HICSS, Hawaii,* 2002.
7. P.-J. Wan, K. M. Alzoubi, and O. Frieder, “Distributed Construction of Connected Dominating Set in Wireless Ad Hoc Networks,” in *Proceedings of INFOCOM,* 2002.
8. Y. Wang and X.-Y. Li, “Geometric Spanners for Wireless Ad Hoc Networks,” in *Proceedings of 22nd IEEE International Conference on Distributed Computing Systems (ICDCS),* 2002.
9. X.-Y. Li, “Algorithmic, Geometric and Graphs Issues in Wireless Networks,” Survey paper in WCMC, 2002.
10. C. E. Perkins and E. M. Royer, “Ad-Hoc on Demand Distance Vector routing,” in *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, New Orleans,* pp. 90–100, February 1999.
11. J. Broch, D. Johnson, and D. Maltz, “The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks,” 1998.
12. C. E. Perkins and P. Bhagwat, “Highly Dynamic Destination-Sequenced Distance-Vector Routing (dsdv) for Mobile Computers,” *Computer Communications Review,* 234–244, October 1994.
13. M. Joa-Ng and I-T. Lu, “A Peer-to-Peer Zone-Based Two-Level Link State Routing for Mobile Ad Hoc Networks,” *IEEE Journal on Selected Areas in Communication, 17,* 8, 1415–1425, August 1999.
14. Vincent D. Park and M. Scott Corson, “A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks,” in *Proceedings of IEEE INFOCOM,* 1997.
15. E. Royer and C. Toh, “A Review of Current Routing Protocols for Ad-Hoc Mobile Wireless Networks,” *IEEE Personal Communications,* April 1999.
16. P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia, “Routing with Guaranteed Delivery in Ad Hoc Wireless Networks,” *ACM/Kluwer Wireless Networks, 7,* 6, 609–616, 2001; *Third International Workshop on Discrete Algorithms and methods for mobile computing and communications,* 1999, 48–55.
17. B. Karp and H. T. Kung, “Gpsr: Greedy Perimeter Stateless Routing for Wireless Networks,” in *ACM/IEEE International Conference on Mobile Computing and Networking,* 2000.
18. M. Mauve, J. Widmer, and H. Harenstein, “A Survey on Position-Based Routing in Mobile Ad Hoc Networks,” *IEEE Network Magazine, 15,* 6, 30–39, 2001.
19. I. Stojmenovic and X. Lin, “Power-Aware Localized Routing in Wireless Networks,” in *IEEE International Parallel and Distributed Processing Symposium,* 2000.
20. M. Bahramgiri, M. T. Hajiaghayi, and 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),* 2002, pp. 392–397.
21. M. Grünewald, T. Lukovszki, C. Schindelhauer, and K. Volbert, “Distributed Maintenance of Resource Efficient Wireless Network Topologies,” 2002, Submitted for publication.

22. L. Hu, "Topology Control for Multihop Packet Radio Networks," *IEEE Trans. Communications, 41,* 10, 1993.

23. Li Li, Joseph Y. Halpern, Paramvir Bahl, Yi-Min Wang, and Roger 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.

24. L. Lloyd, Rui Liu, Madhav V. Marathe, Ram Ramanathan, and S. S. Ravi, "Algorithmic Adpects of Topology Control Problems for Ad Hoc Networks," in *IEEE MOBIHOC,* 2002.

25. S. Narayanaswamy, V. Kawadia, R. Sreenivas, and P. Kumar, "Power Control in Ad-Hoc Networks: Theory, Architecture, Algorithm and Implementation of the Compow Protocol," in *European Wireless Conference,* 2002.

26. R. Ramanathan and R. Rosales-Hain, "Topology Control of Multihop Wireless Networks Using Transmit Power Adjustment," in *IEEE INFOCOM,* 2000.

27. Y.-C. Tseng, Y.-N. Chang, and B.-H. Tzeng, "Energy-Efficient Topology Control for Wireless Ad Hoc Sensor Networks," in *Proceedings of International Conference on Parallel and Distributed Systems (ICPADS),* 2002.

28. R. Wattenhofer, L. Li, P. Bahl, and Y.-M. Wang, "Distributed Topology Control for Wireless Multihop Ad-Hoc Networks," in *IEEE INFOCOM'01,* 2001.

29. R. Rajaraman, "Topology Control and Routing in Ad Hoc Networks: A survey," *SIGACT News, 33,* 60–73, 2002.

30. Godfried T. Toussaint, "The Relative Neighborhood Graph of a Finite Planar Set," *Pattern Recognition, 12,* 4, 261–268, 1980.

31. K. R. Gabriel and R. R. Sokal, "A New Statistical Approach to Geographic Variation Analysis," *Systematic Zoology, 18,* 259–278, 1969.

32. P. Bose, L. Devroye, W. Evans, and D. Kirkpatrick, "On the Spanning Ratio of Gabriel Graphs and Beta-Skeletons," in *Proceedings of the Latin American Theoretical Infocomatics (LATIN),* 2002.

33. X.-Y. Li, P.-J. Wan, and Y. Wang, "Power Efficient and Sparse Spanner for Wireless Ad Hoc Networks," in *IEEE International Conference on Computer Communications and Networks (ICCCN01),* pp. 564–567, 2001.

34. T. Lukovszki, *New Results on Geometric Spanners and Their Applications,* Ph. D. thesis, University of Paderborn, 1999.

35. X.-Y. Li, P.-J. Wan, Y. Wang, and O. Frieder, "Sparse Power Efficient Topology for Wireless Networks," in *IEEE Hawaii International Conference on System Sciences (HICSS),* 2002.

36. S. Arya, G. Das, D. Mount, J. Salowe, and M. Smid, "Euclidean Spanners: Short, Thin, and Lanky," in *Proceedings of 27th ACM STOC,* pp. 489–498, 1995.

37. Y. Wang and X.-Y. Li, "Distributed Spanner with Bounded Degree for Wireless Ad Hoc Networks," in *International Parallel and Distributed Processing Symposium: Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing,* April 2002.

38. H. B. Hunt III, M. V. Marathe, V. Radhakrishnan, S. S. Ravi, D. J. Rosenkrantz, and R. E. Stearns, "NC-Approximation Schemes for NP- and PSPACE-Hard Problems for Geometric Graphs," *Journal of Algorithms, 26,* 2, 238–274, 1999.

39. X.-Y. Li, I. Stojmenovic, and Y. Wang, "Partial Delaunay Triangulation and Degree Limited Localized Bluetooth Multihop Scatternet Formation," 2002, Submitted for publication. The short version appeared at AdHocNow, 2002.

40. S. Datta, I. Stojmenovic, and J. Wu, "Internal Node and Shortcut Based Routing with Guaranteed Delivery in Wireless Networks," *Cluster Computing, 5,* 2, 169–178, 2002.

41. I. Stojmenovic and S. Datta, "Power and Cost Aware Localized Routing with Guaranteed Delivery in Wireless Networks," in *Proc. Seventh IEEE Symposium on Computers and Communications ISCC,* 2002.

42. M. Seddigh, J. Solano Gonzalez, and I. Stojmenovic, “Rng and Internal Node Based Broadcasting Algorithms for Wireless One-to-One Networks,” *ACM Mobile Computing and Communications Review, 5,* 2, 37–44, 2002.

43. D. P. Dobkin, S. J. Friedman, and K. J. Supowit, “Delaunay Graphs are Almost as Good as Complete Graphs,” *Discr. Comp. Geom.,* 399–407, 1990.

44. J. M. Keil and C. A. Gutwin, “The Delaunay Triangulation Closely Approximates the Complete Euclidean Graph,” in *Proc. First Workshop Algorithms Data Structure (LNCS 382),* 1989.

45. J. M. Keil and C. A. Gutwin, “Classes of Graphs which Approximate the Complete Euclidean Graph,” *Discr. Comp. Geom., 7,* pp. 13–28, 1992.

46. X.-Y. Li, G. Călinescu, and P.-J. Wan, “Distributed Construction of Planar Spanner and Routing for Ad Hoc Wireless Networks,” in *21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM),* vol. 3, 2002.

47. G. Călinescu, “Computing 2-Hop Neighborhoods in Ad Hoc Wireless Networks,” submitted for publication.

48. J. Gao, L. J. Guibas, J. Hershburger, L. Zhang, and 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.

49. X.-Y. Li and Y. Wang, “Localized Construction of Bounded Degree Planar Spanner for Wireless Ad Hoc Networks,” submitted for publication.

50. J. Monks, V. Bharghavan, and W.-M Hwu, “Transmission Power Control for Aultiple Access Wireless Packet Networks,” in *IEEE Conference on Local Computer Networks (LCN),* 2000.

51. M. Sanchez, P. Manzoni, and Z. Haas, “Determination of Critical Transmission Range in Ad-Hoc Networks,” in *Multiaccess, Mobility and Teletraffic for Wireless Communications (MMT’99),* 1999.

52. M. Faloutsos and M. Molle, “Creating Otimal Distributed Algorithms for Minimum Spanning Trees,” Tech. Rep. Technical Report CSRI-327 (also submitted in WDAG ’95), 1995.

53. R. Gallager, P. Humblet, and P. Spira, “A Distributed Algorithm for Minimumweight Spanning Trees,” *ACM Transactions on Programming Languages and Systems, 5,* 1, 66–77, 1983.

54. J. A. Garay, S. Kutten, and D. Peleg, “A Sub-Linear Time Distributed Algorithm for Minimum-Weight Spanning Trees,” in *Symposium on Theory of Computing,* 1993, pp. 659–668.

55. L. Kirousis, E. Kranakis, D. Krizanc, and A. Pelc, “Power Consumption in Packet Radio Networks,” in *Symposium on Theoretical Aspects of Computer Science (STACS) ’97,* 1997.

56. A. E. F. Clementi, P. Penna, and R. Silvestri, “On the Power Assignment Problem in Radio Networks,” 2000.

57. A. Clementi, P. Penna, and R. Silvestri, “The Power Range Assignment Problem in Radio Networks on the Plane,” in *XVII Symposium on Theoretical Aspects of Computer Science (STACS’00), LNCS(1770),* pp. 651–660, 2000.

58. A. Clementi, P. Penna, and R. Silvestri, “Hardness Results for the Power Range Assignment Problem in Packet Radio Networks,” in *II International Workshop on Approximation Algorithms for Combinatorial Optimization Problems (RANDOM/APPROX’99), LNCS(1671),* pp. 197–208, 1999.

59. G. Călinescu, I. Mandoiu, and A. Zelikovsky, “Symmetric Connectivity with Minimum Power Consumption in Radio Networks,” in *IFIP-TCS,* 2002, To appear.

60. P. Bose, A. Brodnik, S Carlsson, E. D. Demaine, R. Fleischer, A. Lopez-Ortiz, P. Morin, and J. I. Munro, “Online Routing in Convex Subdivisions,” in *International Symposium on Algorithms and Computation,* pp. 47–59, 2002.

61. P. Bose and P. Morin, “Online Routing in Triangulations,” in *Proc. of the 10th Annual International Symposium on Algorithms and Computation ISAAC,* 1999.

62. I. Stojmenovic and X. Lin, "Loop-Free Hybrid Single-Path/Flooding Routing aAgorithms with guaranteed delivery for Wireless Networks," *IEEE Transactions on Parallel and Distributed Systems, 12,* 10, 2001.

63. S. Basagni, I. Chlamtac, V. R. Syrotiuk, and B. A. Woodward, "A Distance Routing Effect Algorithm for Mobility (DREAM)," in *Proceedings of ACM/IEEE MobiCom'98,* 1998.

64. Z. Haas and B. Liang, "Ad-hoc Mobility Management with Uniform Quorum Systems," *IEEE/ACM Transactions on Networking, 7,* 2, pp. 228–240, 1999.

65. I. Stojmenovic, "A Routing Strategy and Quorum Based Location Update Scheme for Ad Hoc Wireless Networks," Tech. Rep. TR-99-09, Computer Science, SITE, University of Ottawa, 1999.

66. K. N. Amouris, S. Papavassiliou, and M. Li, "A Position Based Multi-Zone Routing Protocol for Wide Area Mobile Ad-Hoc Networks," in *Proceedings of 49th IEEE Vehicular Technology Conference,* pp. 1365–1369, 1999.

67. E. Kranakis, H. Singh, and J. Urrutia, "Compass Routing on Geometric Networks," in *Proceedings of 11 th Canadian Conference on Computational Geometry,* pp. 51–54, 1999.

68. P. Morin, *Online Routing in Geometric Graphs,* Ph.D. thesis, Carleton University School of Computer Science, 2001.

69. P. Bose and P. Morin, "Competitive Online Routing in Geometric Graphs," in *Proceedings of the VIII International Colloquium on Structural Information and Communication Complexity (SIROCCO 2001),* pp. 35–44, 1999.

70. X.-Y. Li, Y. Wang and O. Frieder, "Localized Routing for Wireless Ad Hoc Networks," *2003, IEEE ICC 2003,* to appear.

71. M. Penrose, "The Longest Edge of the Random Minimal Spanning Tree," *Annals of Applied Probability, 7,* 340–361, 1997.

72. P. Gupta and P. R. Kumar, "Critical Power for Asymptotic Connectivity in Wireless Networks," in *Stochastic Analysis, Control, Optimization and Applications: A Volume in Honor of W. H. Fleming,* W. M. McEneaney, G. Yin, and Q. Zhang (eds.), 1998.

73. X.-Y. Li, Y. Wang, P.-J. Wan and C.-W. Yi, "Fault Tolerant Deployment and Topology Control in Wireless Networks," in *2003 ACM MobiHoc,* 2003.

74. M. Penrose, "On k-Connectivity for a Geometric Random Graph," *Random Structures and Algorithms, 15,* 145–164, 1999.

75. B. Bollobás, *Random Graphs,* Cambridge University Press, 2001.

76. H. Dette and N. Henze, "Some Peculiar Boundary Phenomena for Extremes of *r*th Nearest Neighbor Links," *Statistics & Probability Letters, 10,* 381–390, 1990.

77. M. Penrose, "Extremes for the Minimal Spanning Tree on Normally Distributed Points," *Advances in Applied Probability, 30,* 628–639, 1998.

78. M. Penrose, "A Strong Law for the Longest Edge of the Minimal Spanning Tree," *Annals of Probability, 27,* 246–260, 1999.

79. C. Bettstetter, "On the Minimum Node Degree and Connectivity of a Wireless Multihop Network," in *3rd ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'02),* June 2002.

80. D. M. Blough, M. Leoncini, G. Resta, and P. Santi, "On the Symmetric Range Assignment Problem in Wireless Ad Hoc Networks," in *Proceedings of 2nd IFIP International Conference on Theoretical Computer Science,* 2002.

81. C. Cooper and A. Frieze, "On the Connectivity of Random *k*-th Nearest Neighbour Graphs," *Combinatorics, Probability and Computing, 4,* 343–362, 1995.

82. M. Grossglauser and D. Tse, "Mobility Increases the Capacity of Ad-Hoc Wireless Networks," in *INFOCOMM,* 2001, vol. 3, pp. 1360–1369.

83. P. Gupta and P. Kumar, “Capacity of Wireless Networks,” Techincal Report, University of Illinois, Urbana–Champaign, 1999.

84. O. D. Patrick, “Connectivity in Ad-Hoc and Hybrid Networks,” in *IEEE INFOCOM,* 2002.

85. P. Santi and D. M. Blough, “An Evaluation of Connectivity in Mobile Wireless Ad Hoc Networks,” in *Proceedings of IEEE DSN,* pp. 89–98, 2002.

86. F. Xue and P. R. Kumar, “The Number of Neighbors Needed for Connectivity of Wireless Networks,” submitted to *Wireless Networks.*

87. P. Hall, “Distribution of Size, Structure and Number of Vacant Regions in a High-Intensity Mosaic,” *Z. Warsch. verw, Gebiete 70,* 237–261, 1985.

88. T. Lukovszki, “New Results of Fault Tolerant Geometric Spanners,” in *Workshop on Algorithms and Data Structures,* pp. 193–204, 1999.

89. F. Kuhn, R. Wattenhofer, and A. Zollinger, “Asymptotically Optimal Geometric Mobile Ad-Hoc Routing,” in *Proceedings of the 6th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIALM),* 2002.

CHAPTER 7

BROADCASTING AND ACTIVITY SCHEDULING IN AD HOC NETWORKS

IVAN STOJMENOVIC and JIE WU

7.1 INTRODUCTION

Wireless networks consist of static or mobile hosts (or nodes) that can communicate with each other over wireless links without any static network interaction. Each mobile host has the capability to communicate directly with other mobile hosts in its vicinity. They can also forward packets destined for other nodes. Examples of such networks are ad hoc, local area, packet radio, and sensor networks. They are used in disaster rescues and wireless conferences, on battlefields, in possibly remote or dangerous environments where monitoring objects is required, for wireless Internet access, and so on.

In a broadcasting task, a source node sends the same message to all the nodes in the network. In the *one-to-all* model, transmission by each node can reach *all* nodes that are within radius distance from it, whereas in the *one-to-one* model, each transmission is directed toward only *one* neighbor (using narrow-beam directional antennas or separate frequencies for each node). In the literature, broadcasting has been studied mainly for the one-to-all model, and most of this chapter is devoted to that model. The *one-to-many* model can also be considered, in which fixed or variable angular-beam antennas can be used to reach several neighbors at once.

Broadcasting is also frequently referred to in literature as *flooding.* Broadcasting applications include paging a particular host or sending an alarm signal. The broadcasting task was sometimes studied in the context of address serving [1] in hierarchically clustered packet radio networks. Flooding/broadcasting is also used for route discovery in source-initiated, on-demand routing. Broadcasting can similarly be used in the context of an efficient location-aware routing algorithm as follows. The source *S* may initiate a destination

Mobile Ad Hoc Networking. Edited by Basagni, Conti, Giordano, and Stojmenovic.
ISBN 0-471-37313-3 © 2004 Institute of Electrical and Electronics Engineers, Inc.

search process by broadcasting a short message that contains the location of *S*, identity (ID) of destination *D*, and some control bits. When the destination search message successfully reaches *D*, *D* applies any location-based routing algorithm (e.g., [2] which guarantees delivery if the location of the destination, in this case *S*, is accurate) and reports back to *S* with a short message containing its location. The source *S* can then again apply the same routing algorithm [2] (or use the path created in the previous step by *D* if that path was recorded in the process) to send the full message to *D*. Ho et al. [3] argued that flooding can be a viable candidate for multicast and routing protocols in very dynamic ad hoc networks.

Another application of broadcasting is in the sensor network. Recent advances in technology have made it possible to integrate microsensor, low-power signal processing, computation, and low-cost wireless communication into a sensing device, such as one developed by the WINS project at UCLA [4]. Data broadcasting and gathering are important functions supported in a sensor network to collect and disseminate critical information, such as temperature, pressure, and noise level.

Geocasting is a form of broadcasting in which nodes that shall receive messages are restricted to be inside a region. A simple solution to this problem is to route from the source to a node inside the geo-casting region, and then apply broadcasting inside the region [5]. Solutions proposed in literature do not appear to be more efficient than this one, and we will not survey them here. A survey is given in [6].

The traditional solution to the broadcasting problem is *blind flooding,* whereby each node receiving the message will retransmit it to all its neighbors. The only "optimization" applied to this solution is that nodes remember messages received for flooding, and do not act when receiving repeated copies of the same message. However, blind flooding causes unnecessary collisions and bandwidth waste, with many nodes not receiving the message as a consequence.

Williams and Camp [7] classified the broadcast protocols into simple (blind) flooding, probability-based, area-based, and neighbor-knowledge methods. In this chapter, area-based methods are reclassified within other groups, whereas neighbor-knowledge methods are divided into clustering-based, selecting forwarding neighbors, and internal-node-based methods. We shall present here a comprehensive taxonomy of broadcasting schemes with the one-to-all model in mind (the other models can similarly be considered). All schemes can be classified following the taxonomy consisting of five categories: determinism, network information, reliability, "hello" message content, and broadcast message content. The boldfaced terms are used in the summary table given in the conclusion section.

7.1.1 Determinism

A broadcast scheme may use **probabilistic** or **deterministic** protocols, based on whether or not a random number selection was used to make decisions. The random-number usage here is limited to the network layer decision; the underlying medium-access control (MAC) protocol may still use random backoff counters, for example, in a network layer deterministic scheme.

7.1.2 Network Information

The second classification is based on the type of state information used in the algorithm: global or local. Note that the distinction between global and local is not clear-cut. Centralized algorithms can be also applied in the distributed setting if a deciding node has full global information for the network. Through several rounds of sequential information exchanges, global or partial global information can be assembled based on local information only. However, sequential information propagation (also called chain reaction) could be costly, and this can be measured in terms of rounds. Mobility adds another dimension of complexity in measuring state information. The locality of maintenance can be used to measure the adaptiveness of a protocol in the mobile environment. Wu and Lou [8] further classified protocols based on neighbor-knowledge information: **global, quasiglobal, quasilocal,** and **local.** The *global* broadcast protocol, centralized or distributed, is based on global state information. A survey of centralized broadcasting algorithms (using global information) is given in [9], and they will not be covered in this chapter. The classical approximation algorithm by Guha and Khuller [10] for connected dominating sets is based on global information. In *quasiglobal* broadcasting, a broadcast protocol is based on partial global state information. For example, the approximation algorithm in [11] is based on building a global spanning tree (a form of partial global state information) that is constructed in a sequence of sequential propagations. Recently, Chen and Liestman [12] proposed a distributed formation of a weakly connected dominating set by iteratively expanding and connecting fragments, similar to the distributed Kruskal's algorithm. In *quasilocal* broadcasting, a distributed broadcast protocol is based on mainly local state information and occasional partial global state information. Cluster networks are examples of this: although clusters can be constructed locally most of the time, the chain reaction does occur occasionally. In *local* broadcasting, a distributed broadcast protocol is based solely on local state information. All protocols that select forward nodes locally (based on one-hop or two-hop neighbor sets) belong to this category. It has been recognized that *scalability* in wireless networks cannot be achieved by relying on solutions in which each node requires global knowledge about the network. To achieve scalability, the concept of *localized* algorithms was proposed. These algorithms, based on local knowledge, achieve a desired global objective.

7.1.3 Reliability

Reliability is the ability of a broadcast protocol to reach all the nodes in the network. It can be considered at the network layer or at the medium-access layer. We will classify protocols according to their network layer performance. That is, assuming that the MAC layer is ideal (every message sent by a node reaches all its neighbors), a location update protocol provides accurate desired neighborhood information to all nodes, and the network is connected, broadcast protocols can be **reliable** or **unreliable.** In a *reliable* protocol, every node in the network is reached. The set of nodes that rebroadcast a message in a reliable broadcasting scheme define a connected dominating set. A dominating set $D(S)$ of a set S is a set of nodes such that each node from S either belongs to $D(S)$ or has a neighboring node that belongs to $D(S)$. It is easy to observe that all nodes will receive the message if it is retransmitted only by nodes that belong to a connected dominating set. Connectivity provides propagation through the whole network, whereas domination assures reachability by all nodes. The broadcasting task can therefore be solved optimally by finding a connected dominating set of minimal size. Optimality here is measured by the percentage of saved retransmissions in a reliable broadcasting scheme. Unfortunately, the problem of finding a connected dominating set of minimal size is NP-complete, even if a node has global knowledge about the network [13–15]. Therefore one must apply heuristics to flood intelligently. Note also that a protocol, such as blind flooding, that is reliable on the network layer may be very unreliable at the MAC layer. Excess messages in any protocol affect the node power and bandwidth available; thus, the main goal is to describe a reliable broadcast protocol with minimal number of retransmissions, that is, to construct a connected dominating set of minimal size. Note also that the MAC layer cannot guarantee 100% reliability due to the hidden-terminal problem (a node simultaneously receiving messages from two other nodes that are not aware of each other's transmission) and the probabilistic nature of protocols used.

7.1.4 "Hello" Message Content

The broadcast schemes may require different neighborhood information, which is reflected in the contents of messages sent by nodes when they move, react to topological changes, change activity status, or simply periodically send update messages. A commonly seen "hello" message may contain, in addition to its own **ID,** its **position, one bit** for dominating set status (one bit telling neighbors whether or not the node considers itself to be in a dominating set), a list of **one-hop** neighbors, and its **degree** (number of its neighbors). Other content is also possible, such as a list of one-hop neighbors with their positions, a list of two-hop neighbors, or even **global** network information. The Global Position System (GPS) provides geographic location information (if required) to hosts in a wireless network by communication with a satellite network. Alternatively, nodes may measure time delays or signal strengths of incoming messages and determine the relative location of their neighbors.

7.1.5 Broadcast Message Content

The broadcast message sent by the source, or retransmitted, may contain the broadcast **message only.** In addition, it may contain a variety of information needed for proper functioning of the broadcast protocol, such as the information previously listed for "hello" messages, **message plus one/two bits,** or a list of **forwarding neighbors,** informing them whether or not to retransmit the message.

The performance of broadcast protocols can be measured by a variety of metrics. A commonly used metric is the number of message retransmissions (or the total power used in the case of broadcasting with adjusted transmission radii) with respect to the number of nodes (alternatively, *rebroadcast savings,* a complementary measure, can be used). The next important metric is *reachability,* or the ratio of nodes connected to the source that received the broadcast message. Time delay or latency is sometimes used, which is the time needed for the last node to receive the broadcast message initiated at the source. Note that retransmissions at MAC layer are normally deferred, to avoid message collisions. Some authors consider as an alternative a more restricted indicator, whether or not the path from source to any node is always following a shortest path. This measure may be important if used as part of the routing scheme, since route paths are created during the broadcast process.

Intelligent and scalable broadcasting and activity-scheduling solutions are based on the concept of dominating sets. Clusterheads and gateway nodes in a cluster structure define such a set, and were the first "intelligent" flooding solutions proposed in the literature. However, the node mobility either worsens the quality of the structure dramatically, or otherwise causes a chain reaction (local changes in the structure could trigger global updates). Localized connected dominating set concepts, proposed recently, avoid such chain reactions and produce similar or better rebroadcast savings. Their maintenance does not require any communication overhead in addition to maintaining positions of neighboring nodes, or information about two-hop neighbors. One such concept is based on covering all two-hop neighbors by a minimal set of one-hop neighbors. The other is based on creating a fixed dominating set, in which nodes that do not have two unconnected neighbors and nodes that are “covered” by one or two neighbors (each neighbor of a covered node is neighbor of one of nodes that cover it) are eliminated. Neighbor elimination has also been applied (solely or in conjunction with other concepts), whereby nodes give up retransmitting if they are not aware of any neighbor that did not already receive the same message. This chapter will survey known techniques based on dominating sets, and will discuss their advantages and drawbacks.

Ad hoc networks are best modeled by the unit graphs constructed in the following way. Two nodes *A* and *B* are neighbors if and only if the distance between them is at most *R*, where *R* is the transmission radius, and is equal for all nodes. This model is widely used (most protocols in this chapter use it), although many solutions surveyed here are valid in more general models as well.

The remaining sections describe known scalable broadcasting techniques, and compare their performance. Most presented schemes are very recent, developed in the last few years, and the reference list is comprehensive in order to provide fairness to all contributing authors. Sections 7.2–7.6 describe network layer broadcasting schemes using omnidirectional antennas. Section 7.7 discusses the MAC layer for the one-to-all model. Section 7.8 discusses activity scheduling and power-aware broadcasting schemes. Section 7.9 deals with broadcasting based on the use of narrow angular-beam directional antennas. Section 7.10 describes localized schemes for broadcasting with adjusted transmission radii. The Conclusion section provides a table with a summary of broadcasting schemes for the one-to-all communication model, following the presented taxonomy.

7.2 CLUSTERING-BASED FLOODING

The distributed clustering algorithm [16, 17] is initiated at all nodes whose ID is lowest among all their neighbors (locally lowest ID nodes). All nodes are initially undecided. If all neighbors of node *A* that have lower ID sent their cluster decisions and none declared itself a clusterhead (CH), node *A* decides to create its own cluster and broadcasts such decision and its ID as cluster ID. If a node receives a message from a neighbor that announced itself as CH, it will send a message (to all its neighbors) declaring itself a non-CH node, to enable more clusters to be created (note that two CHs are not direct neighbors in the algorithm). Thus, each node broadcasts its clustering decision after all its neighbors with lower IDs have already done so. Non-CH nodes that hear two or more CHs will declare themselves as *gateway* nodes. A sophisticated maintenance procedure for cluster formation when nodes move is described in [17]. To minimize the number of clusters, [18] proposed to apply node degree as the primary key in clusterhead decisions. Nodes with more neighbors are more likely to become clusterheads. In case of ties, lower ID nodes have the advantage. The scheme [11] does not apply degree as the primary key, but instead reduces the number of gateway nodes. After the clustering process is completed, each CH contacts neighboring CHs (up to three hops away) in order to eliminate some gateway nodes, and use only essential gateway nodes to preserve overall connectivity. In Figure

7.1, nodes B and F in the first round create clusters. Then nodes J and C create two more clusters. Nodes A, D, E, and G are gateway nodes. If optimization [11] is applied (based on a spanning-tree maintenance), node A can be eliminated. A clustering-based algorithm is also reported in [19]. It does not depend on any spanning tree, and each node requires knowledge of its single-hop neighbors, and a constant number of two-hop and three-hop neighbors. The construction is fully localized. The maintenance is also localized using an approach similar to that in [20] and outlined below.

Blind flooding has been replaced in [1, 21] by a method in which each CH and gateway node in a clustered wireless network forwards the message exactly once. CHs and gateway nodes together form a connected dominating set. When their scheme is applied, 8 out of 12 nodes need to retransmit the message in Figure 7.1 (or 7 if node A is eliminated). The experiments in [22] gave surprisingly stable ratios of nodes in clustering-based dominating sets, with respect to number of nodes in the network and average node degrees. About 65% of nodes in lowest-ID-based and 52% of nodes in degree-based cluster structures belong to the dominating set. The maintenance of cluster structure, however, requires excessive communication overhead due to the "chain effect" caused by mobility [23, 24]. Although the lowest-ID or highest-node-degree cluster algorithms are localized (with delayed decisions), they has no localized maintenance properties. To achieve localized maintenance, the cluster maintenance can use a different algorithm to make the update a localized one [20]: once the cluster is constructed, a non-CH will never challenge the current CH. If a CH moves into an existing cluster, one of the CHs will give up its role of CH based on some predefined priority. The localized maintenance is preserved, at the price of increasing number of clusters with node mobility.

Gerla, Kwon, and Pei [23] proposed a combined clustering and broadcasting algorithm that has no communication overhead for either maintaining cluster structure or updating neighborhood information. In their passive clustering algorithm, the cluster structure is updated with existing traffic by adding two bits to each ongoing message. The source *S* of a broadcasting task will transmit the message to all its neighbors. *S* will declare itself a CH (for the timeout period that is a parameter in the method) if it has no neighboring active CH. Upon receiving the message, each node *A* will declare itself a CH using the same

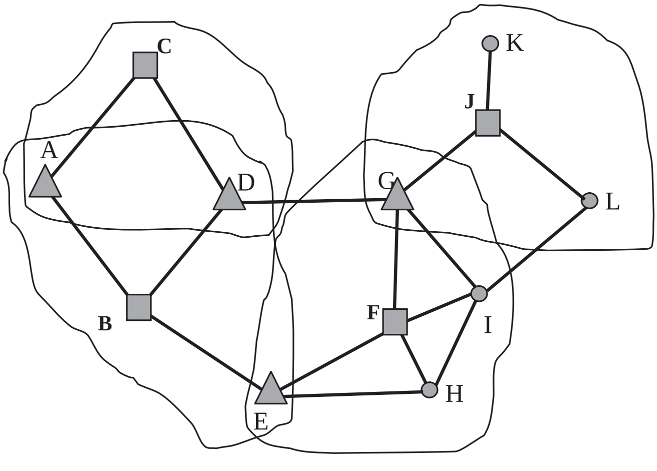


Figure 7.1. Four clusters B, C, F, and J with three or four gateway nodes.

criterion as the source *S*. Otherwise, *A* will check the ratio of neighboring CHs and neighboring gateway nodes and declare itself a gateway if that ratio is above a certain threshold, which is also a parameter of the method. If *A* decides to be a gateway, it will retransmit the message. Otherwise *A* decides to be an ordinary node and does not retransmit the message. The method is not reliable (there are pathological cases of poor delivery ratio) and has global parameters.

To reduce overhead in constructing a connected dominating set among clusterheads, Wu and Lou [8] recently proposed the 2.5-hop coverage, instead of the traditional three-hop coverage (i.e., CHs within three hops) to ensure CH connectivity and full coverage. Instead of using a three-hop coverage area (i.e., CHs within three hops), each CH just covers the CHs that have members (including CHs) within two hops. In Figure 7.1, suppose the network is partitioned to four clusters B (with member E), C (with members A and D), and F (with members G, H, and I), and J (with members K and L). The coverage area of F includes C (which is three hops away) since C's member D is two hops away. The coverage area of B does not include J because none of J's members are within two hops.

7.3 PROBABILISTIC, COUNTER, AND LOCATION BASED SCHEMES

Ni, Tseng, Chen, and Sheu [25] studied the broadcast storm problem. A straightforward broadcasting by flooding is usually very costly and will result in serious redundancy, contention, and collision. They identified this broadcast storm problem by showing how serious it is through analyses and simulations. Several schemes (probabilistic, counter-based, distance-based, location-based, and cluster-based) to reduce redundant rebroadcasts and differentiate timing of rebroadcasts to alleviate this problem are proposed in [25]. These schemes achieve a high percentage of delivery rate with low number of retransmissions. However, they are not reliable. In the probabilistic scheme [25], each node rebroadcasts the first copy of a received message with a given probability *p*. In the counter-based scheme [25], each node rebroadcasts the message if and only if it received the message from less than *C* neighbors. In the distance-based scheme [25], the message is retransmitted if and only if the distance to each neighbor that already retransmitted the message is $>D$. In the location-based scheme [25], the message is retransmitted if and only if the additional area that can be covered if the node rebroadcasts the message (divided by the area of the circle with transmission radius) is greater than the threshold *A*. A simplified version of the method is to rebroadcast the message if the node is not located inside the convex hull of neighboring nodes that already retransmitted the message. In the cluster-based scheme, the lowest-ID clustering algorithm [17] is applied, and one of the above four methods is then applied on CHs and gateway nodes. All described methods are not reliable, and the experimental data [25, 22] indicate low saved rebroadcasts for high reachability.

Sasson, Cavin, and Schiper [26] observe that probabilistic flooding [25] in random unit graphs behaves differently for low- and high-density networks. For low-density networks, the success rate varies linearly with probability, making the method inefficient. For high average degrees, there exists an ideal value of probability, and the success rate drops when it is increased or decreased. Beyond an ideal value, packet collisions become more frequent and network performance degrades.

Cartigny and Simplot [27] described a distance-based method without using position information. The distance between two neighboring nodes is measured by a formula that depends on the number of common neighbors. The broadcast message is piggybacked with a list of one-hop neighbors. Neighbor elimination (see Section 7.6) is also used to enhance the performance. The method is suitable for highly mobile environments, since “hello” message content is minimized.

7.4 SOURCE-DEPENDENT DOMINATING SETS

We shall now present methods that are reliable and fully localized. That is, node mobility impacts only local structure. This section deals with methods in which the selection of forwarding nodes depends on the source of the broadcasting task.

Several authors [14, 15, 28, 29] proposed independently reliable broadcasting schemes in which the sending node selects adjacent nodes that should relay the packet to complete the broadcast. The IDs of selected adjacent nodes are recorded in the packet as a forward list. An adjacent node that is requested to relay the packet again determines the forward list. This process is iterated until broadcast is completed. The methods differ in details on how a node determines its forward list.

The multipoint relaying method, discussed in detail by Qayyum, Viennot, and Laouiti [15], and dominant pruning method, proposed by Lim and Kim [14], are both based on a heuristic that selects a minimal-sized subset of neighbors of a given node S that will “cover” all two hop neighbors of S. A node is called “covered” if it received (directly or via retransmissions by other nodes) the message originating at S. Relay points of S are one-hop neighbors of S that cover all two-hop neighbors of S. That is, after all relay points of S retransmit the message, all two-hop neighbors of S will receive it. The goal is to minimize the number of relay points of S. The computation of a multipoint relay set with minimal size is a NP-complete problem, as proven in [14, 15]. A heuristic algorithm, called the greedy-set cover algorithm, is proposed in [30]. This algorithm repeats selecting node B in which the number of neighbor nodes that are not covered yet is maximized. Consider the network in Figure 7.2, with node F being the source of

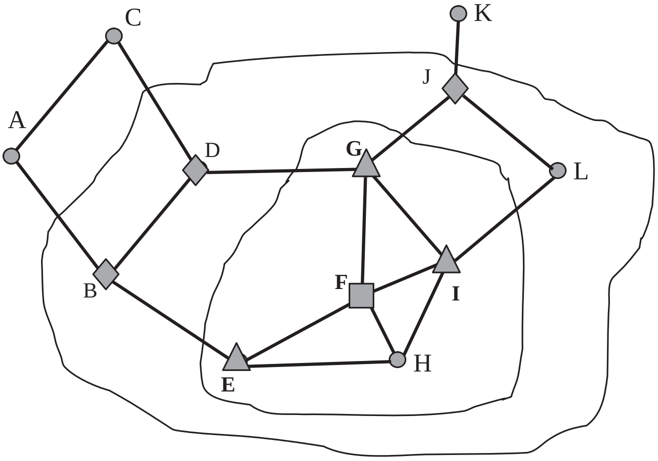


Figure 7.2. Selecting forwarding neighbors: G, E, and I for node F, D, and J for G, and B for E.

broadcasting or a relay node. Its one-hop neighbors are E, G, H, and I, and its two-hop neighbors are B, D, J, and L. E covers only B, G covers D and J, I covers L, and H does not cover any two-hop neighbor. In the first round, node G is selected to forward the packet. Nodes L and B are still not covered. Nodes B and I must be selected to cover them, whereas node H does not need to be in the list. Thus, the forwarding set for node F is {G, E, I}. Each of them then selects its own forward list. They can optimize the selection by ignoring nodes covered by other nodes in the forward list, if they are aware of their neighbors. Node G, for instance, considers one-hop neighbors D and J to forward to B, C, and K (it learns that its two-hop neighbor L is covered by I), and must select both. Node I will not select any forwarding node, whereas node E will select B to cover A and D. In total, 7 out of 12 nodes will rebroadcast. The performance evaluation in [22] gave a quite stable ratio with respect to average graph degree of medium density, with 59–64% of nodes in the dominating set.

Lou and Wu [31] discuss two extended dominant pruning methods: total dominant pruning (TDP) and partial dominant pruning (PDP), both using one-hop neighbors to cover two-hop neighbors. TDP requires that the sender piggyback information about its one-hop and two-hop neighbor sets (simply called neighbor set within two hops) along with the broadcast packet. With this information, the receiver can prune all the nodes in the sender's neighbor set within two hops from the receiver's neighbor set that is within two hops. Apparently, TDP will generate a smaller forward node set than Lim and Kim's dominant pruning (DP), but it also introduces some overhead when the broadcast packet piggybacks the two-hop neighborhood information. PDP, without using the piggybacking technique, directly extracts the neighbors of the common neighbors of both sender and receiver from the receiver's neighbor set within two hops. In Figure 7.2, suppose I is the sender and F is the receiver, then the two-hop neighbor set for I includes D, E, F, G, H, I, J, and L and the two-hop neighbor set for F includes B, D, E, F, G, H, I, J, and L. The coverage set for F is reduced to B based on both TDP and PDP. If F is the sender and E is the receiver, D can be pruned from E's coverage area using TDP, but not for PDP since the link between D and G is not included in E's two-hop neighborhood information. Simulation results in [31] show that the PDP algorithm avoids the extra cost of the TDP algorithm introduced by piggybacking two-hop neighborhood information with the broadcast packet, but achieves almost the same performance improvement.

Note that a pruning approach that is based on neighbor position rather than two-hop neighbor set can also be used [22]. In Figure 7.2, once F determines that the distance between its neighbor G and the incoming node I is less than the transmission radius, there is no need to cover G. However, neighbor position alone is not sufficient to detect neighbors of common neighbors as in PDP. Therefore, neighbor position information only is weaker than information of the neighbor set within two hops.

The extended pruning methods perform well in the average case. However, they do not have a good approximation ratio (the worst-case ratio of selected forward set size with respect to the minimum connected dominating set), especially in a dense network. Wu and Lou [8] propose to extend the pruning method to the cluster network. The extended pruning method is applied to the cluster graph consisting of clusterheads only. Basically, the notion of the cluster graph converts any dense graph to a sparse one to guarantee a constant approximation ratio. The 2.5-hop coverage model is used and it is shown in [8] that the resultant cluster graph is a connected directed graph if the original graph is connected. The authors refer to a version with a localized maintenance property (applying the variantin [20]). However, this may create an excessive number of clusters; thus, cluster-based broadcasting solutions appear to be far from optimal localized solutions for dynamic ad hoc networks.

The adaptation of multihop relaying presented in [32] improves performance in the following manner. The broadcasting node transmits a list of its neighbors at the time of broadcast packet transmission, not as part of any "Hello" message. Two-hop neighbor knowledge is used to determine which neighbors also received the broadcast packet in the same transmission, and these nodes are already covered and are removed from the neighbor graph used to choose the next-hop relaying nodes. Finally, if a broadcast message is received from a node that is not listed as a neighbor, the message is retransmitted, to deal with high mobility issues. In connected dominating set based broadcast algorithm [33], sender node establishes priorities between forwarding nodes, and each forwarding node should eliminate from the consideration not only neighbors of the sender node, but also neighbors of each relaying node with higher priority.

Sun and Lai [29, 34, 35], and Calinescu, Mandoiu, Wan, and Zelikovsky [28] presented heuristics that aimed at covering the whole area where two-hop neighbors could be located by a minimal set of one-hop neighbors, and analyzed the performance of their schemes. The problem is equivalent to selecting a minimal set of disks that still cover the same area as the area covered by all disks centered in neighboring points. Their forwarding sets contain, on average, more nodes than the one based on set cover heuristics [30], since the forwarding sets are given larger areas to cover, and no two-hop information is used. Thus, only one-hop position information is used. The solutions are based on the notion of curved convex hulls, and have sophisticated details that are beyond the scope of this chapter. Each forwarding node in the variants discussed in [29, 28] includes a forwarding set as part of the message. In the variant discussed in [34, 35], this is avoided by transferring the overhead to hello messages, which contain the position of the sender and the list of its neighbors (without position information). The two-hop neighbor information and one-hop position information are used to calculate the local cover set of the sender's node at the receiver's node.

Sisodia, Manoj, and Murthy [36] propose to select forward nodes based on the notion of stability. A weight function that indicates the temporal stability and spatial stability of a node's neighbors is used as the criterion in the selection process.

The lightweight and efficient network-wide broadcast protocol by Sucec and Marsic [37] relies on two-hop neighbor knowledge obtained from "Hello" packets. Each node decides to rebroadcast based on knowledge of which of its other one- and two-hop neighbors are expected to rebroadcast. Neighbors with a high degree of knowledge have higher priority to rebroadcast. Since a node relies on its higher-priority neighbors to rebroadcast, it can proactively compute if all of its lower-priority neighbors will receive those rebroadcasts; if not, the node rebroadcasts.

Rogers [38] proposed a GPS screening-angle technique in which the nodes make the forwarding decision based on the angle between the previous node itself and the next node. If the angle is greater than a threshold value, the message is forwarded to the corresponding neighbor; otherwise, it is not forwarded. Stojmenovic and Seddigh [39] described a method in which each node retransmits the message if and only if it has at least one neighbor further from the source than itself. It will do so also in the case in which a closer neighbor to the source remains silent. The two techniques [38, 39] are not reliable. Boukerche [40] proposed to replace the flooding method in the route discovery phase of the DSR routing algorithm, in which the message is forwarded to each neighbor of any node receiving it, with the GPS screening-angle technique [38] or further-neighbor scheme [39]. It was shown in [40] that occasional failure to discover the destination still causes fewer problems in routing than the extensive overhead of "blind" flooding method that can easily congest the network.

7.5 SOURCE-INDEPENDENT DOMINATING SETS

Most methods presented in the previous section include a forwarding set of neighbors as part of the message. They therefore have message overhead, and the set of retransmitting nodes depends on the source node. The approach presented in this section does not require inclusion of the forwarding set in the message, and has a fixed set of retransmitting nodes, regardless of source choice. Its maintenance does not require more communication overhead, and it offers competitive performance (enhanced with neighbor elimination; see the next section) according to experiments in [22].

Nodes that belong to a (fixed, source-independent) dominating set will be called *internal* nodes (of course, a different definition for the dominating set leads to a different set of internal nodes). It is desirable, in the context of broadcasting, to create a dominating set with minimal possible ratio of internal nodes. Wu and Li [24] proposed a simple and efficient distributed algorithm for calculating a connected dominating set in ad hoc wireless networks. They introduced the concept of an *intermediate* node. A node A is an *intermediate* node if there exist two neighbors, B and C, of A that are not direct neighbors themselves (see Figure 7.3). For example, nodes C and K in Figure 7.3 are not intermediate nodes, while the other nodes are. The concept is simple, but not many nodes are eliminated from the dominating set. If a graph were complete, the definition might be modified to select the highest key node as the default dominating set, although no retransmission is needed for reliable broadcast.

Wu and Li [24] also introduced two rules that considerably reduce the number of internal nodes in the network. Rule 1 [24] is as follows. Consider two intermediate neighboring nodes v and u. If every neighbor of v is also a neighbor of u, and $id(v) < id(u)$, then node v is not an *intergateway* node. We may also say that node v is "covered" by node u. Observe that retransmission by v, in this case, is covered by retransmission of u, since any node that might receive the message from v will receive it instead from u. Stojmenovic et

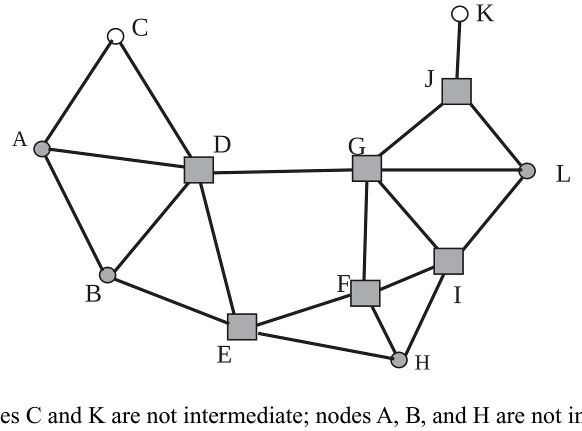


Figure 7.3. Nodes C and K are not intermediate; nodes A, B, and H are not intergateway nodes.

al. [22] proposed to replace node IDs with a record *key* = (*degree*, *x*, *y*), where *degree* is the number of neighbors of a node (and is the primary key in the comparison), and *x* and *y* are its two coordinates in the plane (and serve as secondary and ternary keys). It significantly reduces the size of the dominating set. Using such keys, consider example in Figure 7.3. Note that node J is forced by node K, for whom it is the only neighbor, to be in the dominating set for all possible definitions of dominating sets that do not include node K in it. Nodes A and B are covered by node D, node H is covered by node F, and node L is covered by node G. The remaining six nodes are intergateway nodes, and are shown as squares in Figure 7.3.

Next, let the *gateway* nodes be those intergateway nodes that are not eliminated by Rule 2 [24], defined as follows. Assume that *u*, *v*, and *w* are three intergateway nodes that are mutual neighbors. If each neighbor of *v* is a neighbor of *u* or *w*, where *u* and *w* are two connected neighbors of *v*, and *v* has the lowest *id* among the three, then *v* can be eliminated from the list of gateway nodes. Stojmenovic et al. [22] again proposed to use the above-defined *key* instead of *id*. The reason for the elimination of *v* is that any node that can benefit from retransmission by *v* will receive the same message instead from either *u* or *w*. All intergateway nodes in Figure 7.3 remain gateway nodes. Node E is "covered" by D and F, but D and F are not connected themselves. Although all neighbors of node I are neighbors of either F or G, it does not have lowest *id* (in this example, the *x* coordinate serves as *id*). If *id* is changed appropriately, node I may become covered. This suggests that further improvements to the gateway definition might be possible, but the enhancement may require informing neighbors about dominating set status. In the current definition, nodes may decide their own dominating set status without any message exchange, but cannot decide the same for their neighbors.

Stojmenovic et al. [22] observed that covering by one or two nodes may be done by any nodes, not only (inter)gateway ones, leading to the same decision about dominating set status. These neighbors, if not themselves (inter)gateway nodes, will be transitively further covered [eventually by (inter)gageway nodes]. The consequence of this observation [22] is that neighbors do not need to exchange their (inter)gateway status in order to make their own decisions, and therefore the decisions can be made without any message exchanged between neighbors. In our judgment, this is, together with the quality of the structure itself, the most desirable property of a topology construction and maintenance algorithm.

If location information of neighboring nodes is available, each node can determine whether or not it is an intermediate, intergateway, or gateway node in $O(k^3)$ computation time (where *k* is the number of its neighbors), without any message exchanged with its neighbors for that purpose. Otherwise, the maintenance of internal node status requires the knowledge of neighbors for each neighbor. Experiments in [22] indicate that the percentage of gateway nodes decreases from 60% to 45% when average graph degree increases from 4 to 10.

Dai and Wu [41] proposed several enhancements to the definition of internal nodes. In [41], they generalize one- and two-neighbor coverage of a node to *k*-neighbor coverage, with fixed and variable *k*. The case of variable *k* is even computationally less expensive than two-node coverage. In this definition, each node *A* considers the subgraph of its neighboring nodes with higher keys than *A*, and constructs connected components in the subgraph (depth-first search can be used for this task). If there exists one connected component so that each neighbor of *A* is a neighbor of at least one node from the component, then node *A* is not a gateway node. Note that the test can be further simplified by observing that, in order to cover *A*, all neighbors with higher keys must be connected, that is, there must be exactly one connected component.

A source-independent definition of dominating set in applications where the dominating-set status of each node must be communicated to its neighbors (this is the case in routing and activity scheduling applications) can be described as follows [42]. Each node *A* initially calculates its dominating set status based on the original gateway node definition [24]. Using some backoff mechanism, each gateway node decides when to transmit its decision to its neighbors (nongateway nodes remain silent). While waiting, it may hear several announcements from its gateway node neighbors. After each announcement, *A* reevaluates its gateway node decision. If the subgraph of all neighboring nodes with higher key value or with announced gateway node decision is connected, and each neighbor of *A* is a neighbor of at least one of these nodes, then *A* decides to withdraw from the dominating set and never transmits such decision to neighbors.

A two-hop dominating-set concept, which can be further generalized to a *k*-hop dominating-set concept, and can be viewed as a clustering scheme with localized maintenance property, is proposed in [43]. It has the following properties: Two neighboring clusterheads can be at distance one or two; each node in a cluster is at distance one or two from its clusterhead; and two clusters are, thus, connected if there exists a node that is neighbor to both clusterheads, or the two clusterheads are directly linked. The structure is a generalization of Wu's dominating-set concept [24]. We shall define similarly two-hop intermediate, intergateway, and gateway nodes as follows: A node *X* is two-hop intermediate if it has two two-hop neighbors *B* and *C* that are not two-hop neighbors themselves. A two-hop intermediate node *X* is a two-hop intergateway if it has no two-hop neighbor *Y* such that every two-hop neighbor of *X* is also a two-hop neighbor of *Y*, and $key(X) < key(Y)$. The value $key(X)$ can be one of $id(X)$, $[degree(X), id(X)]$, or $[energy\text{-}level(X), degree(X), id(X)]$, that is, it can have primary, secondary, ternary, and so on keys. for comparisons. A two-hop inter-gateway node *X* is a two-hop gateway if it has no two neighbors *Y* and *Z* such that every two-hop neighbor of *X* is a two-hop neighbor of *Y* or *Z*, and $key(X) < key(Y)$, $key(X) < key(Z)$. Adjih, Jacquet, and Viennot [44] proposed to combine multipoint relay and dominating-set approaches. Each node computes its forwarding neighbor's set and transmits it to its neighbors. Each node then determines whether it belongs to a "MPR-dominating set" if it either has the smallest ID in its neighborhood, or the node is a forwarding neighbor of the neighbor with the smallest ID.

7.6 NEIGHBOR ELIMINATION

A neighbor-elimination scheme has been independently proposed in four papers [14, 22, 39, 45]. In this source-dependent scheme, a node does not need to rebroadcast a message if all its neighbors have been covered by previous transmissions. In order to apply the method, the same assumption as in the previous section is taken: either nodes learn geographic positions of their neighbors, or receive a list of neighbors from each of their neighbors. After each received copy of the same message, the node eliminates from its rebroadcast list neighbors that are assumed to receive correctly the same message. If the list becomes empty before the node decides to rebroadcast, the rebroadcasting is canceled. The neighbor-elimination scheme version from [45] uses two-hop neighbor information instead of location of one-hop neighbors.

The method depends on the selected medium-access scheme. If IEEE 802.11 is used,

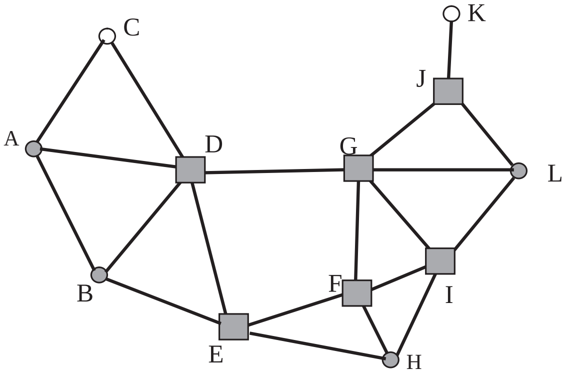


Figure 7.4. Retransmitting nodes (square shaped) in a neighbor-elimination method with source A.

Peng and Li [45] propose to let nodes with more neighbors rebroadcast earlier, so that more nodes can be covered by one transmission, but experiments in [22] did not find significant differences from the scheme in which nodes choose backoff times at random within a fixed interval. Consider, for example, the network in Figure 7.4. Let us assume that the order of retransmissions corresponds to the *x*-coordinate, that is, proceeds from left to right. Node A is the source. Node B retransmits, followed by node C, which is not aware that node D already received the message from B. Retransmissions from D, E, F, G, and J then follow. Node I, for instance, does not retransmit since all its neighbors are covered by previous retransmissions from F and G. Note that there exist some additional retransmissions with respect to the gateway node set, but also some gateway nodes (e.g., I) may not need to retransmit.

Although this neighbor-elimination scheme alone was not competitive with other dominating set definitions, it was able to improve the performance of all of them as an added feature. For instance, if a gateway node (e.g., I in Figure 7.4) realizes that all its neighbors are covered by previous transmissions, it will not rebroadcast. Further, in [22], it was found that each noninternal node A will assign itself to a neighboring internal node B that has the largest degree. In case of ties, use the lowest *id* among candidate neighbors. This rule attaches more neighbors to higher-degree nodes, thus possibly "emptying" the assigned list of low-degree internal nodes. If both internal node status and neighbor-elimination schemes are applied, then the algorithm works as follows: When an internal node receives a message, it retransmits the message if it has a noneliminated neighboring node that is either a noninternal node assigned to it, or an internal node. Similar enhancements can be made to multipoint relay, and all methods come from [25]. Better methods (like gateway-dominating set) can be improved by about 1%, whereas others benefit up to 10% in saved rebroadcasts.

Wu and Dai [46] described a scheme that can be viewed as a combination of a generalized internal-node-based dominating set [24] and neighbor elimination, generalized to several last hops in the broadcast path. In that scheme, gateway nodes are defined on the fly, and the status may depend on the source node. The subgraph consists of all *k*-hop neighbors (for small *k* such as 1 or 2) of A with higher priority value (say, *id*), and all *k*-hop neighboring nodes that have previously forwarded the message (if the routing history up to certain number of hops is included in the packet). If there exists a connected component of that subgraph so that any neighbor of A is a neighbor of at least one node from the component then A is a nongateway node. A more generalized rule is also proposed in [24]: A node A is nongateway node if every pair of its neighbors is connected via nodes with either higher priority value or nodes that have forwarded the message.

Cartigny, Ingelrest, and Simplot [47] described a RNG relay subset scheme, in which a node *v* retransmits the message received from *u* if *v* has an RNG (the concept is described below) neighbor which is not covered by *u*'s transmission. The algorithm can be interpreted as the neighbor-elimination method whereby each node immediately eliminates all its non-RNG neighbors from its forward list. After this preliminary step, each node behaves as in the neighbor-elimination scheme. Euclidean and neigborhood-based distances are considered.

7.7 RELIABLE BROADCASTING

In this section, we shall discuss the broadcasting problem at the medium-access layer. The design of a reliable broadcast method depends on the following three decisions [48]:

1. By whom are errors detected?
2. How are error messages signaled?
3. How are missing packets re-transmitted?

Decisions (1) and (2) are normally handled jointly.

In the sender-initiated approach, the sender is responsible for the error detection. Error messages are signaled using ACK signals sent from each receiver. Missing data at a receiver is detected if the sender does not receive an ACK from the receiver. In this case, missing packets are retransmitted from the source through a unicast. When several receivers have missing packets, the sender may decide to rebroadcast the missing packets to all the receivers.

In the receiver-initiated approach, each receiver is responsible for the error detection. Instead of acknowledging each broadcast packet, each receiver sends a NACK once it detects a missing packet. Suppose broadcast packets are time stamped using a sequence number; a missing packet can then be detected by a gap between sequence numbers of the receiving packets.

When the sender-initiated approach is applied, only the sender (which keeps the history of broadcast packets) is responsible for retransmitting the missing packet, and the corresponding retransmitting method is called sender-oriented. Note that when the sender receives ACK signals from all the receivers, the corresponding packet can be removed from the history.

There are three ways to retransmit the missing packet when the receiver-initiated approach is used: sender-oriented, neighborhood-oriented, and fixed-neighborhood-oriented. These methods differ by the locations of the copies of missing packets. These locations are also called copy sites; they include the sender. Note that when there are several receivers that have the same missing packet, broadcast NACK signals will be sent to the copy site(s). To ensure that at most one NACK is returned to the sender per packet transmission, when a receiver detects a missing error, it waits a random period of time before broadcasting a NACK to the sender and all other receivers. This process is called NACK suppression since a receiver will cancel its broadcast if it receives a NACK that corresponds to a packet it has missed. In the sender-oriented approach, senders can either unicast to a receiver that needs the missing packet or broadcast to all the receivers. In the neighborhood-oriented approach, the receiver that needs the missing packet searches its neighborhood for a group member that keeps a copy of the missing packet. The search process uses a TTL-based unicast process or TTL-based broadcast process. The search space is either limited to the broadcast tree (which is now rooted at the receiver) or is without limitation. In the fixed-neighborhood-oriented approach, the copy sites are fixed to a subgroup or each receiver has a "buddy" to back up each other.

Mobility of mobile ad hoc networks adds complexity in achieving reliability. When a host moves from one neighborhood to another, proper hand-off protocols are needed. For example, when host *U* has just completed its forwarding process to its neighbor *V*, host *W*, a neighbor of *V*, moves away from the neighborhood of *V* and enters the neighborhood of *U*. To ensure that host *W* gets a copy of the packet, *U* needs to keep the copy for a while and will reforward the packet (with a proper tag indicating that this is a reforwarding packet) whenever a change in its neighborhood is detected.

In [31], Lou and Wu study two environments to handle mobility. In the "static" environment, mobile hosts are allowed to roam freely in the working space. However, the broadcast process (including forward-node selection and the broadcast process itself) is done quickly so that both one-hop and two-hop neighbor sets remain the same during the process for each host. In addition, each host has updated and consistent one-hop and two-hop neighbor sets when the broadcast process starts. Clearly, delivery of the broadcast packet is guaranteed as long as the selected forward nodes cover all hosts. In the "dynamic" environment, the broadcast process is still done quickly as in the static environment, so that both one-hop and two-hop neighbor sets remain the same during the process for each host. However, a host cannot update its one-hop and two-hop neighbor sets in a timely and consistent manner because mobile hosts are moving at a fast speed. Under this model, the broadcast delivery rate is no longer 100%. A simulation result in [31] shows that the broadcast delivery rate still remains high in an ad hoc network with slow- to moderate-speed mobile hosts (with respect to the transmission range) using an ideal MAC layer without contention and collision. This high delivery rate is partly due to the broadcast redundancy in selecting the forwarding nodes. Therefore, although excessive broadcast redundancy is harmful and will cause the broadcast storm problem, some degree of redundancy is useful for reliability purposes.

Hsu, Tseng, and Sheu [49] propose an efficient reliable broadcast protocol based on end-to-end acknowledgment, that is, all acknowledgments will be sent back to the source following the reverse of the broadcast tree. The tree is constructed redundantly—each node has multiple parent nodes (one primary and several backups). However, if all parent nodes are lost (due to the movement of hosts), flooding is needed to guarantee that all the acknowledgments will eventually be sent to the source.

Pagani and Rossi [21] propose a two-level hierarchical scheme for reliable broadcast. Two phases are used: scattering and gathering. In the scattering phase, the broadcast packet is forwarded to all clusterheads, which, in turn, send it to their local members. In the gathering phase, the acknowledgments are collected by each clusterhead and sent along the broadcast tree, built on the clusterheads, back to the source.

In order to approach 100% reachability rate in an IEEE 802.11 environment, Stojmenovic et al. [22] designed the RANA (Retransmission After Negative Acknowledgements) broadcasting algorithm. When a node *A* re-transmits a message, and if a collision at receiving node *B* occurs before the sender is recognized, no retransmission request is issued. If the collision occurred after recognizing the sender node *A*, but before receiving the full message, *B* will send a negative acknowledgment to *A*, asking it to repeat the transmission. The reachability in [22] improved from over 94% to over 98%, but with a trade-off (up to 10% more retransmissions). The hidden-terminal problem (two nonneighboring nodes receiving a message simultaneously and rebroadcasting it to a common neighbor) is the main obstacle to achieving 100% reliability in a network operating in IEEE 802.11 medium-access scheme.

Viswanath and Obraczka [50] proposed different heuristics to deal with broadcast reliability in highly mobile environments. Based on local movement velocity, each node decides between three modes for the broadcasting task. In scoped flooding [50], periodic "hello" messages contain a one-hop neighbor list. If the receiving node's neighbor list is a subset of the transmitting node's list, then it does not re-broadcast the packet. We note that this is a special case of the neighbor-elimination scheme [14, 22, 45]. The plain flooding mode is the same as blind flooding. In the hyperflooding mode, additional rebroadcasts can be triggered upon receiving a packet from a new neighbor.

7.8 ACTIVITY SCHEDULING AND POWER-AWARE BROADCASTING

In ad hoc wireless networks, the limitation on the power of each host poses a unique challenge for power-aware design [51]. There has been an increasing focus on low-cost and reduced-node power consumption in ad hoc wireless networks. Even in standard networks such as IEEE 802.11, requirements are included to sacrifice performance in favor of reduced power consumption. In order to prolong the life span of each node and, hence, the network, power consumption should be minimized and balanced among nodes. Unfortunately, nodes in the dominating set in general consume more energy in handling various bypass traffic than nodes outside the set. Therefore, a static selection of dominating nodes will result in a shorter life span for certain nodes, which in turn will result in a shorter life span of the whole network.

Wu, Wu, and Stojmenovic [52] study dynamic selection of dominating nodes, also called activity scheduling. Activity scheduling deals with the way to rotate the role of each node among a set of given operation modes. For example, one set of operation modes is sending, receiving, idle, and sleeping. Different modes have different energy consumptions. Activity scheduling judiciously assigns a mode to each node to save overall energy consumption in the networks and/or to prolong life span of each individual node. Note that saving overall energy consumption does not necessarily prolong the life span of a particular individual node. Specifically, they propose to save overall energy consumption by allowing only dominating nodes (i.e., gateway nodes) to retransmit the broadcast packet. In addition, in order to maximize the lifetime of all nodes, an activity scheduling method is used that dynamically selects nodes to form a connected dominating set. Specifically, in the selection process of a gateway node, preference is given to a node with a higher energy level. The effectiveness of the proposed method in prolonging the life span of the network is confirmed through simulation. Source-dependent forwarding sets appear to be more energy balanced. However, it was experimentally confirmed in [53] that the difference in energy consumption between an idle node and a transmitting node is not major, whereas the major difference exists between the idle and sleep states of nodes. Therefore the most energy efficient methods will select a static dominating set for a given round, turning all remaining nodes to a sleep state. Depending on energy left, changes in activity status for the next round will be made. The change can, therefore, be triggered by changes of power status in addition to node mobility. From this point of view, internal node-based dominating sets provide static selection for a given round and more energy efficiency than the forwarding-set-based method, which requires all nodes to remain active in all the rounds. In [67], the key for deciding dominating set status is a combination of remaining energy and node degree.

Xu, Heidemann, and Estrin [54] discuss the following sensor sleep node schedule. The trade-off between network lifetime and density for this cell-based schedule was investigated in [68]. The given two-dimensional space is partitioned into a set of squares (called cells), such as any node within a square can directly communicate with any nodes in an adjacent square. Therefore, one representative node from each cell is sufficient. To prolong the life span of each node, nodes in the cell are selected in an alternative fashion as a representative. The adjacent squares form a two-dimensional grid and the broadcast process becomes trivial. Note that the selected nodes in [54] make a dominating set, but the size of it is far from optimal, and it also depends on the selected size of the squares. On the other hand, the dominating-set concept used here has smaller size and is chosen without using any parameter (the size of the square has to be carefully selected and propagated with relative node positioning in the solution [54]).

The Span algorithm [55] selects some nodes as coordinators. These nodes form a dominating set. A node becomes a coordinator if it discovers that two of its neighbors cannot communicate with each other directly or through one or two existing coordinators. Also, a node should withdraw if every pair of its neighbors can reach each other directly or via some other coordinators (they can also withdraw if each pair of neighbors is connected via possibly noncoordinating nodes, to provide the chance for other nodes to become coordinators). Since coordinators are not necessarily neighbors, three-hop neighboring topology knowledge is required. However, the energy and bandwidth required for maintenance of three-hop neighborhood information is not taken into account in experiments [55]. On the other hand, if the coordinators are restricted to be neighboring nodes, then the dominating-set definition [55] becomes equivalent to the one given by Wu and Li [24]. In addition, the protocol [55] relies heavily on proactive periodic beacons for synchronization, even if there is no pending traffic or node movement. Recent research on energy consumption [53] indicates that the use of such periodic beacons or “hello” messages is an energy-expensive mechanism, because of significant start-up costs for sending short messages. Finally, Chen et al. [55] observed that the overhead required for coordination with SPAN tends to “explode” with node density, and thus counterbalances the potential savings achieved by the increased density.

Feeney [56] described a power-saving protocol in which each station is awake a bit over half the time, to ensure that awake periods of any two neighboring stations will overlap, allowing communication between them.

Tian and Georganas [57] considered a somewhat related problem, the area coverage, in which sensors decide about their activity status to prolong network lifetime but still provide continued monitoring of the whole area assigned. In their solution, nodes observe that their monitoring area is already covered by other active sensors, send a message announcing their withdrawal from monitoring status, and move to passive state. An alternative method [42] follows a dominating-set-based approach in which nodes instead announce their activity status by one added bit; the method is used for both area coverage or dominating-set creation with reduced size of the forwarding node set.

7.9 BROADCASTING WITH DIRECTIONAL ANTENNAS

We shall now discuss the case of broadcasting in the one-to-one model, corresponding to narrow-beam directional antennas. A broadcasting algorithm called SPIN for sending a message from a node in a sensor network to all other nodes is described in [58]. Each node that receives the datum (i.e., the message) that is being broadcast will forward a corresponding *metadatum* that has considerably shorter bit length (e.g., 16 bytes instead of 500) to all its neighbors. Sensor's *id*, message *id*, or sensor's location are examples. The metadatum is thus flooded. The actual datum could be information that a particular sensor collects. Neighboring nodes that did not yet receive the metadatum will reply with a request to get the actual datum. The node will respond by sending the actual datum to all nodes that requested it. The power consumed by the SPIN protocol [58] is $(n-1)E + 2E'$, where n is the number of nodes, e is the number of edges in the graph, and E and E' are mean powers consumed for sending long and short messages, respectively, along one hop. In any broadcast scenario, the energy $(n-1)E$ consumed is inevitable and is a lower bound that needs to be utilized.

An improved broadcasting scheme is described in [59] and is based on the relative neighborhood graph (RNG) concept [60] defined as follows. An edge (u, v) exists between vertices u and v if the distance between them, $d(u, v)$, is not strictly the largest side in any triangle uvw for every common neighbor w of u and v. In other words, $\forall w \neq u,v$: $d(u, v) \leq \max\,[d(u, w), d(v, w)]$. Thus, for an edge (u, v) to be included, the intersection of two circles centered at u and v and with diameter uv in Figure 7.5 (shaded area) should not contain any vertex w from the set. In Figure 7.2, uv is not in RNG because of witness node w. Figure 7.6 shows the RNG on a set of six nodes (in this case the RNG is a spanning tree, which is not always the case).

Toussaint [60] proved several important properties of relative neighborhood graphs, which are necessary for their application in the broadcasting task. RNG is a connected planar graph. The planarity of the graph assures that it is a sparse graph. Each node has on average about 2.5 neighbors independent of unit-graph density.

Thus, we will only try to optimize the number of short messages. In the flooding algorithm [58], the number of short messages is equal to the total number of edges in the network. A huge reduction in the short-message count can be obtained by applying concepts of RNG graphs and dominating sets, as described in [59]. In the RNG-based broadcast, starting from the source, the message is sent once over each edge of the RNG. Thus instead of $nd/2$ messages in a complete unit graph with average density d, it is sent on about $1.25n$ edges, with a reduction factor of $d/2.5$ over the scheme [58]. The scheme, however, requires the distance information between any two neighboring nodes, which can be obtained from time-delay or signal-strength measurements. If that information is not available, Seddigh et al. [59] propose to apply the dominating-set concept that requires two-hop neighbor information at each node. Since any node in the network has an internal

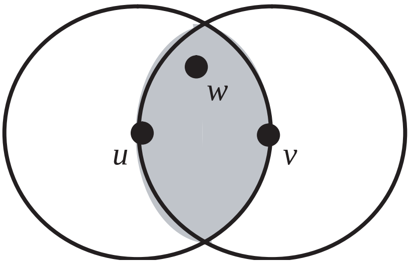


Figure 7.5. (u, v) is not in the RNG graph because of a witness w.

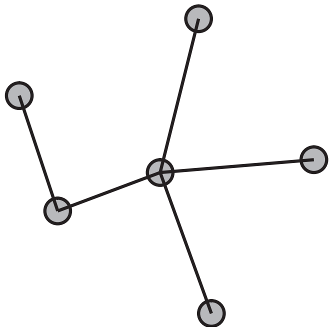


Figure 7.6. RNG graph.

node neighbor, it suffices that only internal nodes retransmit the message. Messages are only sent on edges connecting two internal nodes (one message per edge). The number of short messages is then equal to the number of edges in the subgraph of internal nodes. Each noninternal node, knowing all its internal node neighbors, will choose one of them and inform that one to send all broadcast messages to it. The number of noninternal nodes is therefore added to the number of edges connecting internal nodes. Experiments in [59] show that less than 10% more messages are needed in that approach compared to RNG-based one.

7.10 BROADCASTING WITH ADJUSTED TRANSMISSION RADII

In the minimum-energy broadcasting problem, each node can adjust its transmission power in order to minimize total energy consumption but still enable a message to be originated from a source node to reach all the other nodes in an ad hoc wireless network. The problem is known to be NP-complete [13]. There exist a number of approximate solutions in the literature (cited in [61]) in which each node requires global network information (including distances between any two neighboring nodes in the network) in order to decide its own transmission radius. Cartigny, Simplot, and Stojmenovic [61] described a localized protocol whereby each node requires only the knowledge of its distance to all neighboring nodes and distances between its neighboring nodes (or, alternatively, geographic position of itself and its neighboring nodes). In addition to using only local information, the protocol is shown experimentally to be competitive even with the best-known globalized BIP solution [62], which is a variation of Dijkstra's shortest-path algorithm. The solution [61] is based on the use of a RNG that preserves connectivity and is defined in a localized manner. The transmission range for each node is equal to the distance to its furthest RNG neighbor, excluding the neighbor from which the message came. Localized energy efficient broadcast for wireless networks with directional antennas is described in [63], and is also based on RNG. Messages are sent only along RNG edges, requiring about 50% more energy than the BIP-based [62] globalized solution. However, when the communication overhead for maintenance is added, the localized solution becomes superior.

Lipman, Boustead, and Judge [64] described the following broadcasting protocol.

Table 7.1. Taxonomy of Broadcast Schemes for One-to-All Model with Fixed Transmission Range

Method	Determinism	Network information	Reliability	"Hello" message	Broadcast message
Cluster tree [11]	deterministic	quazi-global	Yes	global	message only
Clustering [1,17, 22]	deterministic	quazi-local	Yes	ID/degree	message only
Passive clustering [23]	deterministic	local	Yes	none	message + two bits
Probabilistic [25]	probabilistic	local	No	ID/degree	message only
Counter/distance, location [25]	probabilistic	local	No	ID/position	message only
Border retransmit [27]	probabilistic	local	No	ID	one-hop
Forwarding neighbors [14, 15, 36]	deterministic	local	Yes	one-hop	forwarding neighbors
Curved convex hull [29, 28]	deterministic	local	Yes	position	forwarding neighbors
Curved convex hull [34, 35]	deterministic	local	Yes	position + one-hop	message only
Forwarding-node cluster [8]	deterministic	local	Yes	ID	forwarding neighbors
Partial dominant prunning [31]	deterministic	local	Yes	one-hop	forwarding neighbors
Lightweight [37]	deterministic	local	Yes	one-hop	message only
MPR-dominating [44]	deterministic	local	Yes	one-hop	message only
Screening angle [38]/ further neighbor [39]	deterministic	local	No	position	message only
Intermediate, Rules 1 & 2 [24]	deterministic	local	Yes	position or one-hop	message only
Intermediate, inter(gateway) [22]	deterministic	local	Yes	[24] + degree	message only
Rule *k*/connected component cover [41]	deterministic	local	Yes	ID	message only
k-hop dominating set [43]	deterministic	local	Yes	$(k-1)$-hop	message only
Announced gateway [42]	deterministic	local	Yes	ID/degree + one bit	message only
k-hop gateway + neighbor elimination [46]	deterministic	local	Yes	$(k-1)$-hop	message + last hops
Neighbor elimination [14, 22, 39, 45]	deterministic	local	Yes	position/ one-hop	message only
Gateway + neighbor elimination [22]	deterministic	local	Yes	position/ one-hop + degree	message only
RNG relay subset + neighbor elim. [47]	deterministic	local	Yes	position/ one-hop + degree	message only

Upon receiving a broadcast message(s) from a node *h*, each node *i* (which was determined by *h* as a forwarding node) determines which of its one-hop neighbors also received the same message. For each of its remaining neighbors *j* (which did not receive a message yet, based on *i*'s knowledge), node *i* determines whether *j* is closer to *i* than any one-hop neighbors of *i* (that are also forwarding nodes of *h*) that received the message already. If so, *i* is responsible for message transmission to *j*, otherwise it is not. Node *i* then determines a transmission range equal to that of the farthest neighbor it is responsible for.

7.11 CONCLUSION

Table 7.1 presents a taxonomy of broadcast protocols for the omnidirectional antenna (one-to-all) model with fixed transmission range, following our discussion. Only network-layer methods are included, that is, MAC layer methods discussed in Section 7.7 are not included. Also not included are a variety of global solutions, which were not within the scope of this chapter.

Despite the rapidly growing number of publications on the broadcasting problem, no comprehensive performance evaluation exists. Williams and Camp [7] compared some selected protocols using the contention-based 802.11 MAC scheme under various network and mobility scenarios. However, they did not include internal-node-based dominating sets [24, 22] in their experiments. The articles that did compare their methods with the internal node-based dominating sets [65, 34] used an inefficient version [24] of it instead of the improved one in [22] (the neighbor-elimination scheme is the main improvement) and have even misunderstood the method, claiming communication overhead for its construction. (In their defense, the description in [24] used "marking process" which is misleading.)

There are some issues not discussed in this survey. For example, Mosko and Garcia-Luna-Aceves [66] considered a series of broadcasting tasks and the impact of such flow on the performance and reliability. Our discussion was restricted to the performance of one broadcast task at a time. They obtained some initial results, and this and other relevant issues will be studied further in literature. Thus, we expect increased research activity on the transport layer of the broadcasting problem.

REFERENCES

1. G. Lauer, "Address Servers in Hierarchical Networks," in *Proceedings of ICC,* pp. 443–451, 1988.
2. P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia, "Routing with Guaranteed Delivery in Ad Hoc Wireless Networks," in *Proceedings of 3rd International Workshop DIAL M,* Seattle, August 20, 1999, pp. 48–55; *ACM/Kluwer Wireless Networks, 7,* 6, 609–616, 2001.
3. C. Ho, K. Obraczka, G. Tsudik, and K. Viswanath, "Flooding for Reliable Multicast in Multihop Ad Hoc Networks," in *Proceedings of 3rd International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communication DIAL'M,* August 1999.
4. G. J. Pottie and W. J. Kaiser, "Wireless Integrated Network Sensors," *Communications of ACM, 43,* 5, 51–58, 2000.
5. I. Stojmenovic, "Voronoi Diagram and Convex Hull Based Geocasting and Routing in Ad Hoc Wireless Networks," in *Proceedings of IEEE International Symposium on Computers and Communications ISCC,* Turkey, July 2003.

6. X. Jiang and T. Camp, “A Review of Geocasting Protocols for a Mobile Ad Hoc Network,” Colorado School of Mines, 2002.

7. B. Williams and T. Camp, “Comparison of Broadcasting Techniques for Mobile Ad Hoc Networks,” *Proceedings of MobiHoc,* Lausanne, Switzerland, June 2002.

8. J. Wu and W. Lou, “Forward-Node-Set-Based Broadcast in Clustered Mobile Ad Hoc Networks,” *Wireless Communications and Mobile Computing, 3,* 2, 155–173.

9. A. Pelc, “Broadcasting in Wireless Networks,” in *Handbook of Wireless Networks and Mobile Computing,* Wiley, pp. 509–528, 2002.

10. S. Guha and S. Khuller, “Approximation Algorithms for Connected Dominating Sets,” Algorithmica, 20, 4, 374–387, 1998.

11. K. M. Alzoubi, P. J. Wan and O. Frieder, “New distributed Algorithm for Connected Dominating set in Wireless Ad Hoc Networks,” in *Proceedings of Hawaii International Conf. System Sciences,* 2002.

12. Y. P. Chen and L. Liestman, “Approximating Minimum Size Weakly-Connected Dominating Sets for Clustering Mobile Ad Hoc Networks,” in *Proceedings of MobiHoc,* 2002.

13. N. F. Huang and T. H. Huang, “On the Complexity of Some Arborescences Finding Problems on a Multi-hop Radio Network,” *BIT, 29,* 212–216, 1989.

14. H. Lim and C. Kim, “Flooding in Wireless Ad Hoc Networks,” in *Proceedings of ACM MSWiM Workshop at MobiCom,* Aug. 2000; *Computer Communication Journal, 24,* 3–4, 353–363, 2001.

15. A. Qayyum, L. Viennot, and A. Laouiti, “Multipoint Relaying: An Efficient Technique for Flooding in Mobile Wireless Networks,” in *Proceedings of Hawaii International Conf. System Sciences,* January 2002.

16. A. Ephremides, J. E. Wieselthier and D. J. Baker, “A Design Concept for Reliable Mobile Radio Networks with Frequency Hoping Signaling,” *Proceedings of IEEE, 75,* 56–73, 1987.

17. C. R. Lin and M. Gerla, “Adaptive Clustering for Mobile Wireless Networks,” *IEEE Journal on Selected Areas in Communications, 15,* 7, 1265–1275, 1997.

18. G. Chen, F. Garcia, J. Solano, and I. Stojmenovic, “Connectivity Cased k-hop Clustering in Wireless Networks,” in *CD Proceedings of Hawaii International Conf. System Science,* January 2002, INIB03.

19. K. M. Alzoubi, P. J. Wan, and O. Frieder, “Message-optimal Connected Dominating set in Mobile Ad Hoc Networks,” in *Proceedings of MobiHoc,* 2002.

20. C. C. Chiang, H. K. Wu, W. Liu, and M. Gerla, “Routing in Clustered Multihop, Mobile Wireless Networks with Fading Ahannel,” in *Proceedings of IEEE Singapore International Conf. on Networks,* pp. 197–211, 1996.

21. E. Pagani and G. P. Rossi, “Providing Reliable and Fault Tolerant Broadcast Delivery in Mobile Ad-Hoc Networks,” *Mobile Networks and Applications, 4,* 175–192, 1999.

22. I. Stojmenovic, M. Seddigh, and J. Zunic, “Dominating Sets and Neighbor Elimination Based Broadcasting Algorithms in Wireless Networks,” *IEEE Transactions on Parallel and Distributed Systems, 13,* 1, 14–25, 2002.

23. M. Gerla, T. J. Kwon, and G. Pei, “On Demand Routing in Large Ad Hoc Wireless Networks with Passive Clustering,” in *Proceedings of IEEE WCNC,* September 2000.

24. J. Wu and H. Li, “A Dominating Set Based Routing Scheme in Ad Hoc Wireless Networks,” in *Proceedings of DIAL M,* Seattle, August 1999, pp. 7–14; *Telecommunication Systems, 18,* 1–2, 13–36, 2001.

25. S.Y. Ni, Y. C. Tseng, Y. S. Chen, and J. P. Sheu, “The Broadcast Storm Problem in a Mobile Ad Hoc Network,” in *Proceedings of MobiCom,* Seattle, August 1999, pp. 151–162.

26. Y. Sasson, D. Cavin, and A. Schiper, “Probabilistic Broadcast for Flooding in Wireless Mobile Ad Hoc Networks,” *Technical Report IC/2002/54,* EPFL, July 2002, *www.epfl.ch.*

27. J. Cartigny, and D. Simplot, "Border Node Retransmission Based Probabilistic Broadcast Protocols in Ad Hoc Networks," *Telecommunication Systems, 22,* 1–4, 189–204.

28. G. Calinescu, I. Mandoiu, P. J. Wan, and A. Zelikovsky, "Selecting Forwarding Neighbors in Wireless Ad Hoc Networks," in *Proceedings of DIAL M,* 2001.

29. M. T. Sun and T. H. Lai, "Location Aided Broadcast in Wireless Ad Hoc Network Systems," in *Proceedings of IEEE Symposium on Ad Hoc Wireless Networks,* at GLOBECOM, November 2001.

30. L. Lovasz, "On the Ratio of Optimal Integral and Fractional Covers," *Discrete Mathematics, 13,* 383–390, 1975.

31. W. Lou and J. Wu, "On Reducing Broadcast Redundancy in Ad Hoc Wireless Networks," *IEEE Transactions on Mobile Computing, 1,* 2, 111–122, 2002.

32. W. Peng and X. Lu, "AHBP: An Efficient Broadcast Protocol for Mobile Ad Hoc Networks," *Journal of Science and Technology,* Bejing, China, 2002.

33. W. Peng and X. Lu, "Efficient Broadcast in Mobile Ad Hoc Networks Using Connected Dominating Sets," in *Proceedings of ICPADS,* Iwate, Japan, July 2000.

34. M. T. Sun and T. H. Lai, "Computing Optimal Local Cover Set for Broadcasting in Ad Hoc Networks," in *Proceedings of IEEE ICC,* pp. 3291–3295, 2002.

35. M. T. Sun and T. H. Lai, "Location Aided Broadcast in Wireless Ad Hoc Network Systems," in *Proceedings of IEEE WCMC,* 2002.

36. R. Sisodia, B. Manoj, and C. Murthy, "A preferred Link Based Routing Protocol for Wireless Ad Hoc Networks," *IEEE/KICS Journal of Communication Networks, 4,* 1, 14–21, 2002.

37. J. Sucec and I. Marsic, "An Efficient Distributed Network-wide Broadcast Algorithm for Mobile Ad Hoc Networks," *CAIP Technical Report 248,* Rutgers University, September 2000.

38. S. Rogers, "GPS Query Optimization in Ad Hoc Mobile Network Routing," 2001.

39. I. Stojmenovic and M. Seddigh, "Broadcasting Algorithms in Wireless Networks," in *Proceedings of International Conf. on Advances in Infrastructure for Electronic Business, Science, and Education on the Internet SSGRR,* L'Aquila, Italy, July 31–Aug. 6, 2000.

40. A. Boukerche, "A Performance Evaluation of a Dynamic Source Routing discovery Optimization Using GPS System," *Telecommunication Systems, 22,* 1–4, 337–354.

41. F. Dai and J. Wu, "Distributed Dominant Pruning in Ad Hoc Wireless Networks," in *IEEE Globecom,* 2002.

42. J. Carle, D. Simplot, and I. Stojmenovic, "Sensor Area Coverage and Activity Scheduling with Reduced Dominating Set Size," in preparation.

43. I. Stojmenovic, "Clustering with Localized Maintenance via 2-hop Dominating Sets," in preparation.

44. C. Adjih, P. Jacquet, and L. Viennot, "Computing Connected Dominating sets with Multipoint relays," *Rapport #4597,* INRIA, October 2002.

45. W. Peng, X -C. Lu, "On the Reduction of Broadcast Redundancy in Mobile Ad Hoc Networks," in *Proceedings of First Annual Workshop on Mobile and Ad Hoc Networking and Computing,* Boston, August 11, 2000, pp. 129–130.

46. J. Wu and F. Dai, "Broadcasting in Ad Hoc Networks Based on Self-pruning," in *Proceedings of IEEE INFOCOM 2003; International Journal of Foundations of Computer Science,* to appear.

47. J. Cartigny, F. Ingelrest, and D. Simplot, "RNG Relay Subset Flooding Protocols in Mobile Ad Hoc Networks," *International Journal of Foundations of Computer Science, 14,* 2, April 2003, 253–266.

48. D. G. Petitt, "Reliable Multicast Protocol Design Choices," in *Proceedings of MILCOM 97,* Vol. 1, November 1997, pp. 242–246.

49. C. S. Hsu, Y. C. Tseng, and J. P. Sheu, "An Efficient Reliable Broadcasting Protocol for Wireless Mobile Ad Hoc Networks," 2002.

50. K. Viswanath and K. Obraczka, "An Adaptive Approach to Group Communications in Multi-

hop Ad Hoc Networks," in *Proceedings of IEEE International Symposium on Computers and Communications ISCC,* Taormina, Italy, July 2002, pp. 559–566.

51. C. Rohl, H. Woesner, and A. Wolisz, "A Short Look on Power Saving Mechanisms in the Wireless LAN Standard Draft IEEE 802.11," in *Proceedings of 6th WINLAB Workshop on Third Generation Wireless Systems,* 1997.
52. J. Wu, B. Wu, and I. Stojmenovic, "Power-aware Broadcasting and Activity Scheduling in Ad Hoc Wireless Networks Using Connected Dominating Sets," in *Proceedings of IASTED WOC 02, July 2002; Wireless Communications and Mobile Computing, 3,* 4, 425–438, June 2003.
53. L. M. Feeney and M. Nilsson, "Investigating the Energy Consumption of a Wireless Network Interface in an Ad Hoc Networking Environment," in *Proceedings of IEEE INFOCOM,* 2001.
54. Y. Xu, J. Heidemann, and D. Estrin, "Geography-informed Energy Conservation for Ad Hoc Networks," in *Proceedings of MobiCom,* 2001.
55. B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, "Span: An Energy-Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks," in *Proceedings of ACM MobiCom,* 2001.
56. L. M. Feeney, "A QoS Aware Power Save Protocol for Wireless Ad Hoc Networks," in *Proceedings of Mediterranean Workshop on Ad Hoc Networks Med-Hoc,* Sardinia, Italy, Sept., 2002.
57. Di Tian and N. Georganas, "A Node Scheduling Scheme for Energy Conservation in Large Wireless Sensor Networks," *Wireless Networks and Mobile Computing, 3,* 2, 271–290, 2003.
58. W. R. Heinzelman, J. Kulik, and H. Balakrishnan, "Adaptive Protocols for Information Dissemination in Wireless Sensor Networks," in *Proceedings of ACM MobiCom,* Seattle, pp. 174–185, 1999.
59. M. Seddigh, J. Solano Gonzalez, and I. Stojmenovic, "RNG and Internal Node Based Broadcasting Algorithms for Wireless One-to-One Networks," *ACM Mobile Computing and Communications Review, 5,* 2, 37–44, 2001.
60. G. Toussaint, "The Relative Neighborhood Graph of a Finite Planar Set," *Pattern Recognition, 12,* 4, 261–268, 1980.
61. J. Cartigny, D. Simplot, I. Stojmenovic, "Localized Energy Efficient Broadcast for Wireless Networks," in *Proceedings of IEEE INFOCOM,* 2003.
62. J. E. Wieselthier, G. D. Nguyen, and A. Ephremides, "On the Construction of Energy-Efficient Broadcast and Multicast Trees in Wireless Networks," in *Proceedings of IEEE INFOCOM,* Tel Aviv, Israel, pp. 585–594., 2000
63. J. Cartigny, D. Simplot, and I. Stojmenovic, "Localized Energy Efficient Broadcast for Wireless Networks with Directional Antennas," in *Proceedings of Mediterranean Workshop on Ad Hoc Networks Med-Hoc,* Sardinia, Italy, Sept., 2002.
64. J. Lipman, P. Boustead, and J. Judge, "Efficient and Scalable Information Dissemination in Mobile Ad Hoc Networks," in *Proceedings of the First International Conf. on Ad-hoc Networks and Wireless ADHOC-NOW,* Toronto, September 2002, pp. 119–134.
65. P. Jacquet, A. Laouiti, P. Minet, and L. Viennot, "Performance of Multipoint Relaying in Ad Hoc Mobile Routing Protocols," in *Proceedings of IFIP Networking 2002,* Pisa, Italy, pp. 387–398, May 2002.
66. M. Mosko and J. J. Garcia-Luna-Aceves, "Performance of Group Communication over Ad Hoc Networks," in *Proceedings of IEEE International Symposium on Computers and Communications ISCC,* Taormina, Italy, July 2002, pp. 545–552.
67. J. Shaikh, J. Solano, I. Stojmenovic, and J. Wu, "New Metrics for Dominating Set Based Energy Efficient Activity Scheduling in Ad Hoc Networks," in *Proceedings of WLN Workshop at IEEE Conference on Local Computer Networks,* Bonn, Germany, October 20–24, 2003.
68. D. M. Blough and P. Santi, "Investigating Upper Bounds on Network Lifetime Extension for Cell-Based Energy Conservation Techniques in Stationary Ad Hoc Networks," in *Proceedings of ACM MobiCom,* Atlanta, Sept. 2002.

CHAPTER 8

LOCATION DISCOVERY

ANDREAS SAVVIDES and MANI B. SRIVASTAVA

8.1 INTRODUCTION

Location discovery is a broad topic that has received considerable attention from the research community during the past few decades. Its significance stems from its wide spectrum of applications ranging from navigation to context-aware applications in ubiquitous computing. As expected, the location requirements vary across applications, resulting in the development of a rich and diverse set of technologies and location discovery algorithms. In cellular telephony systems, for example, knowledge of handset locations is primarily required for routing calls to mobile users. Such location is found either by observing the received signal strength of a broadcast signal transmitted by the base stations or by measuring the round-trip signal propagation time between a handset and a set of base stations.

Although cell-level localization granularity is sufficient for routing calls to mobile handset users, new applications require more precise handset locations. The main motivator for these applications was the 1996 Federal Communications Commission (FCC) second-phase mandate for E911 (enhanced 911) services, which required that by October 1st, 2001 all cellular carriers should be able locate phones that make 911 emergency calls within 50–100 meters, in most cases either by triangulating the positions of handsets that used the received signals or by using GPS [15]. Although the FCC requirements where not entirely met in October 2001 and many extensions and waivers have been granted by the FCC, the E911 mandate has spawned an entirely new industry for location-aware applications, formally named location-based services (LBS). Since then, LBS has become a rapidly growing market and, according to the ARC Group (http://www.arcgroup.com), will grow to U.S. $11 billion in North America, U.S. $13 billion in Asia Pacific, and U.S. $12 billion in Europe.

Mobile Ad Hoc Networking. Edited by Basagni, Conti, Giordano, and Stojmenovic.
ISBN 0-471-37313-3 © 2004 Institute of Electrical and Electronics Engineers, Inc.

The background infrastructure development for LBS services over cellular networks is well underway. A notable example is the Qualcomm gpsOne chipset, which provides low-cost GPS-based technologies with 5 to 10 meter accuracy outdoors and 20 meter accuracy in indoor and suburban environments, making it the key enabler for numerous LBS applications. Another two companies in the United States, TeleCommunication Systems Inc. (TCS) and LocatioNet, have recently created the messaging and middleware infrastructures for developing LBS applications such as point-of-interest (i.e., where are the closest ATM machines, restaurants, movie theaters, etc.), driving directions, and navigation, friend-finder applications (are any of my friends or family nearby?) and mobile commerce applications (are there any deals today in this mall?), yellow page services, and child safety services.

Besides the cellular technologies and LBS, location discovery is the key enabler for a plethora of applications across different domains ranging from navigation systems to virtual reality and augmented systems. Asset and personnel tracking, access control, location-specific billing services, automated inventory control, power control in smart buildings, camera motion tracking, various security applications, and user-context transfer according to user location are just a few of the applications that will revolutionize a whole new era of ubiquitous computing and communication. For the great majority of applications, location discovery is mutually coupled with networking technologies. Location-aware applications require networking support to function, whereas networking protocols and applications, on the other hand, can greatly benefit from utilizing knowledge of location.

8.1.1 The Impact of Location Discovery on Ad Hoc Networks

In ad hoc wireless networks and wireless sensor networks, location discovery plays a major role in the development of geographic-aware routing and multicasting protocols that result in new more efficient ways for routing data in multihop networks that span large geographic regions. Furthermore, with the inexorable progress of wireless and MEMS technologies, and the ever-expanding application space of wireless sensor networks, location discovery is becoming an indispensable component for establishing correspondence between the Internet and the physical world.

Geographic routing protocols such as GPSR, described in [17], use geographic information at the node level to make localized decisions based on the current position of a forwarding node and the geographic location of the destination. This localized decision process relaxes the requirements for maintaining massive amounts of state information at each node, and reduces the amount of traffic generated during the route discovery process, thus enabling network scalability to large number of nodes spanning large regions. The core component of such geographic routing protocols is a geographic forwarding algorithm, which decides the next-hop destination of a packet based on the coordinates of the forwarding node and knowledge of the geographic location of the destination. If the destination is provided as a particular geographic region, then the packet-forwarding mechanisms are sufficient to deliver the packets to the destination. If the a node address is used as a destination, then the packet-forwarding mechanisms also require the support of location services such as those described in [19] to establish the correspondence between a node address and the current geographical location of the node. A detailed survey of geographic-aware protocols is provided in [22].

Geographic multicasting protocols make use of location information in a similar manner to control-packet flooding to user-specified regions of the network. Furthermore, in wireless sensor networks, knowledge of sensor locations is required to correctly report the origins of events when setting alarms, to assist with target tracking, to evaluate network coverage, and to perform different network maintenance tasks (e.g., replace the batteries on specific sensor nodes).

The remainder of this chapter is organized as follows: The first half of the chapter provides a brief background on location discovery and surveys the existing location discovery systems. The second half of the chapter focuses on the recent developments in ad hoc location discovery.

8.2 LOCATION DISCOVERY OVERVIEW

8.2.1 What is Location?

The term “location” is used in many contexts to denote the position of an object in physical space with respect to a specific frame of reference that varies across applications. In the case of the Global Positioning System (GPS), absolute location is provided in terms of latitude, longitude, and altitude. In the terrestrial counterpart of GPS, the LORAN system, locations are usually given with respect to fixed beacon locations. In some applications that also use inertial sensors such as accelerometers, magnetometers, odometers, and gyroscopes, locations are usually referenced with respect to a starting point, whereas in indoor systems, locations are usually provided according a local coordinate system that represents the dimensions of an area of interest or the building coordinates.

Location discovery is fundamentally based on two main phases: a *measurement phase* that produces a set of measurements of distance, angularly or optically to/from a set of anchor points; and a *combining phase* that combines the measurements to produce a final estimate. The granularity of the derived locations depends on the accuracy of the measurement technologies used and the sophistication of the algorithms employed to combine the measurements. For many applications, where cost and simplicity are the dominating decision factors, the determination of proximity is adequate, whereas in other applications such as robotic navigation systems and augmented reality applications, fine-grained locations of objects with centimeter-level accuracies are required.

8.2.2 Measurement Technologies

Measurements are made using different techniques that leverage known characteristics of signal propagation. The most common measurement methods used in location discovery systems are received signal strength, time-based, and directional methods and they are typically applied to radio, acoustic, or optical signals. Inertial measurement is also a popular measurement method, especially in the fields of mobile robotics and augmented reality systems.

Signal-strength-based methods make use of signal attenuation with distance to determine proximity. These methods are not used for accurate distance measurements because of the large variations in signal attenuation in different environments, especially when multipath and shadowing effects are present. Instead, signal-strength-based methods are used to determine proximity or are combined with other methods to determine locations. Olivetti’s active badge system [1] is based on infrared base stations deployed in each room. Each active badge in a certain room can determine if it can receive an infrared signal from that room. RFID tags work in a similar manner using strategically placed tag readers. In this case, the locations of tags can be determined relative to their proximity to tag readers. The GPS-less localization system proposed by Bulusu et al. [3] uses signal reception to determine proximity. A node can determine its proximity to be the centroid of the beacons from which it can receive packets. Microsoft's RADAR system [2], for example, uses radio received signal strength on wireless network cards during an offline phase to create signal strength maps of a building based on the receiver signal strength measurements to different base stations placed around the building. With the help of these maps, the system is then able to determine user locations within the building. A similar approach is used in various localization systems that use Bayesian estimation schemes. First, the system is trained by creating a probabilistic model based on received signal strength and then these models are used to estimate user locations.

Time-based methods measure distances by recording the time of flight (ToF) of a signal from the transmitter to the receiver. One type of time-based methods assumes that the receiver and the transmitter are time synchronized, so the ToF of a signal to reach the receiver can be accurately recorded. The GPS system is an example of such a system. In GPS, the satellites transmit a code and each receiver locally generates a replica of the signal. When the signals transmitted from the satellite arrive at the receiver, the receiver compares the received code with the locally generated replica to determine ToF. Since the satellite clocks and the receiver clock are not perfectly time synchronized, in practice the receiver combines an additional satellite reading and simultaneously computes the time drift between the receiver and satellite clocks. Another type of ToF method uses two signals with different propagation speeds. The time difference between the arrivals of the two signals is used to determine distances. An example system that uses this approach is Active Bat from AT\&T Cambridge research labs [30]. Active Bat uses RF signals to synchronize the receiver to the transmitter so that the ToF of an ultrasonic signal can be accurately recorded. Finally, a third type of time-based method uses the round-trip time of a signal to determine distance. In such systems, the receiver will transmit back the receiver signal immediately upon reception or right after a deterministic delay period. The transmitter records the roundtrip time of this signal to determine distances. The PinPoint system manufactured by RF Technologies [25] operates in this fashion. Other systems employ ToF methods that measure distances using Ultra Wide Band (UWB) radios [8] and laser ranging systems. Finally, another notable distance measurement technique based on wideband acoustics has been developed by Girod et al. [7]. By encoding the signal and transmitting at different frequencies, this system provides robust distance measurement in the presence of interference.

Directional methods use angle of arrival (AoA) or direction of arrival (DoA) for computing locations. The VHF Omnidirectional Range navigation system (VOR) [34] used in airplane navigation is an example of a direction-based system. The basic principle of operation of the VOR is very simple: the VOR facility transmits two signals at the same time. One signal is constant in all directions, while the other is rotated about the station. The airborne equipment receives both signals, looks (electronically) at the difference between the two signals, and interprets the result as a radial distance from the station. The system proposed in [20] uses RF transmissions from rotating directional antennas to determine the locations of wireless sensor nodes.

8.2.3 Geometric Algorithms for Location Discovery

The location of an object can be computed if the set of measurements to a set of landmarks is known. If the measurements are distances, at least three noncollinear measure-

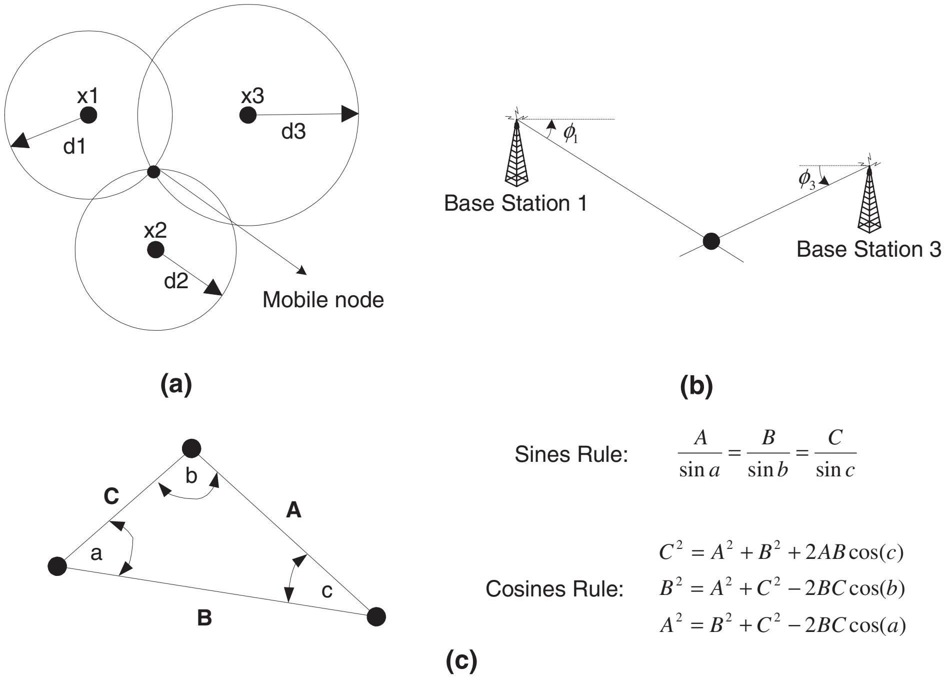

x1
x3
d1
d3
x2
d2
Mobile node
(a)
ϕ_1
Base Station 1
ϕ_3
Base Station 3
(b)
b
C
A
a
c
B
Sines Rule: $\frac{A}{\sin a} = \frac{B}{\sin b} = \frac{C}{\sin c}$
Cosines Rule:
$C^2 = A^2 + B^2 + 2AB\cos(c)$
$B^2 = A^2 + C^2 - 2BC\cos(b)$
$A^2 = B^2 + C^2 - 2BC\cos(a)$
(c)

Figure 8.1. Geometric methods for location discovery.

ments to landmarks with known locations are required to estimate node locations in two-dimensional space (four for three-dimensional space). The location of an object lies at the intersection of three circles with radii equal to the distance between the object and the landmarks (Figure 8.1a). Alternatively, if one is interested in determining location with a local frame of reference, then the location can be easily determined using trigonometric relationships (Figure 8.1b). With angular measurements, position can be determined with only two reference points (Figure 8.1c). The main problem with these methods is that they assume ideal noiseless measurements. In reality, this is not the case, however. Transducers are noisy and several external factors also introduce more sources of error into the measurement system; therefore, the error characteristics of the measurement process need to be carefully considered before computing node locations.

8.3 LOCATION DISCOVERY IN THE PRESENCE OF ERRORS

Location estimation from noisy measurements can be improved if the nature of the error is known. The types of errors depend on the types of signal used and the surrounding environment.

8.3.1 Sources of Error

Multipath Fading and Shadowing. In radio signal strength measurements, multipath fading and shadowing causes up to 30–40 dB variation over distances on the order of half a wavelength. Furthermore, scattering near the receiver will affect AoA measurements, which create problems even when there is a direct line of sight between the receiver and the transmitter. If ToF methods are used, conventional delay estimators based on correlation are influenced by the presence of multipath fading, which results in a shift in the peak of the correlation.

Nonline-of-Sight (NLOS). This affects the AoA methods when the AoA from a longer path is much different from the true AoA. Additionally, in ToF methods, if the direct path to the receiver is blocked, the NLOS component will make the objects appear further than they actually are.

Multiple-Access Interference. This is genrally a problem in CDMA systems in which high-power users may mask the low-power users due to near–far effects. It can also be a problem in acoustic and ultrasonic systems if transmissions from nearby beacon nodes collide.

Transducer Calibration Issues. This is the case in systems that use the received signal strength indicator (RSSI) from low-cost radios to determine range measurements. Since these systems do not use high-precision precalibrated components, they tend to exhibit significant variation in actual transmit power for the same transmit power level or in the RSSI measured for the same received signal strength.

Fluctuations in Signal Propagation Speeds. This error is common in acoustic measurements in which the speed of propagation can be greatly affected by external factors such as wind and fluctuations in temperature and humidity levels. The errors increase over long-distance measurements in which the speed of propagation can vary in different segments of the path. Based on the above errors, a vector of noisy measurements can be considered as an approximation vector of the real measurement of element X:

$$X = \mu(\alpha) + E \tag{8.1}$$

where $\mu(\alpha)$ is the noiseless measurement vector and E is a random vector with known probability density function and covariance matrix σ (see [23] for a more detailed description). The system-measurement model depends on the measurement method used. Two methods commonly used to deal with the measurement discrepancies when estimating locations in the presence of errors are least squares (LS) estimation and Bayesian estimation. The former is used when the measurement error distribution is white gaussian, whereas the latter is used when the error distribution is not gaussian. In this chapter, the examples illustrate the use of least squares methods for estimating node locations. For example location discovery systems that use Bayesian methods, we refer the reader to [5, 18].

8.3.2 Atomic Multilateration

One of the most basic scenarios for estimating location is the one that uses distance measurements to a set of landmarks or beacons. As soon as the measurements are completed, location is estimated by minimizing the sum of the squares of the residuals between measured distances and the corresponding distances derived using the location estimates. We refer to this problem as atomic multilateration, and it can be solved using least squares estimation.

Considering the two-dimensional scenario in Figure 8.2, the residual $f_{u,i}^2$ is given by

$$f_{u,i} = r_{u,i} - \sqrt{(x_i - \hat{x}_u)^2 + (y_i - \hat{y}_u)^2} \tag{8.2}$$

where $r_{u,i}$ is the measured distance between node u with unknown location (x_u, y_u) and beacon i. Location $(\hat{x}_u, \hat{y}_u)$ represents the estimated position of node x, based on a set of measurements. The objective function is to calculate a location estimate for node u such that the residuals between the measured and estimated distances are minimized:

$$F(x_u, y_u) = \min \sum f_{u,i}^2$$

The location of node u in the example scenario can be estimated if at least three distance measurements from the noncollinear, noncolocated beacons are available.

When these conditions are met, the problem becomes overconstrained, resulting in three equations, one for each distance measurement and two unknowns [the (x_u, y_u) coordinates], and the position of node u can be uniquely determined. The solution to this nonlinear optimization problem can be obtained using a suitable gradient-descent method. One possible method is to linearize the equations by taking the Taylor expansion and then applying an iterative minimum mean square estimation (MMSE) solution. The linearized form of the residual from Equation (8.2) is given by

$$r_{i,u} = f_i^{(u)} + \Delta x_i \delta_x + \Delta y_i \delta_y + O(\Delta^2) \tag{8.3}$$

where

$$\Delta x_i = \frac{x_i - \hat{x}_u}{r_i}, \ \Delta y_i = \frac{y_i - \hat{y}_u}{r_i}$$

$$r_i = \sqrt{(x_i - \hat{x}_u)^2 + (y_i - \hat{y}_u)^2}$$

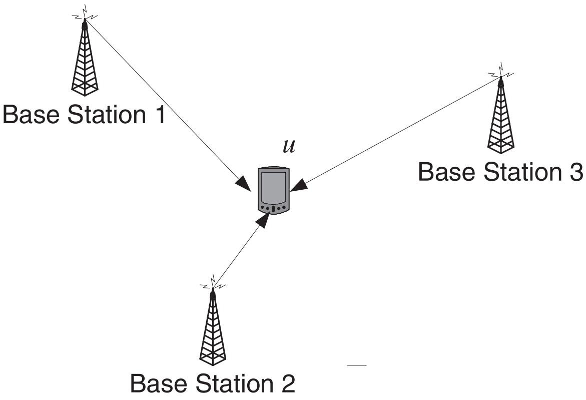


Figure 8.2. Example two-dimensional scenario.

and $O(\Delta^2)$ denotes the higher-order terms that are excluded from the computation. The value of f_i^u is given by $f_i^u = f_i(\hat{x}_u, \hat{y}_u)$, where $\hat{x}_u, \hat{y}_u$ are some initial estimates for x_u, y_u. One possible method for obtaining a suitable initial estimate is provided in Section 8.4.2.8. The equations from these distance measurements can be expressed in matrix form $A\delta = z$ with

$$\delta = \begin{bmatrix} \delta_x \\ \delta_y \end{bmatrix}, A = \begin{bmatrix} \Delta x_1 & \Delta y_1 \\ \Delta x_2 & \Delta y_2 \\ \Delta x_3 & \Delta y_3 \end{bmatrix} \quad \text{and} \quad z = \begin{bmatrix} r_{1,u} - f_1^{(u)} \\ r_{2,u} - f_2^{(u)} \\ r_{3,u} - f_3^{(u)} \end{bmatrix}$$

In this form, one can compute δ using the least squares equation:

$$\delta = (A^T A)^{-1} A^T z$$

in an iterative fashion. At each iteration, δ provides a correction to the position estimate of node u, $\hat{x}_u = \hat{x}_u + \delta_x$ and $\hat{y}_u = \hat{y}_u + \delta_y$. This process is repeated iteratively until δ converges to zero. The computation for weighted least squares is similar and includes the error covariance matrix R, $\delta = [A^T R^{-1} A]^{-1} A^T R^{-1} z$. The associated error covariance matrix Q_0 is given by $Q_0 = [A^T R^{-1} A]^{-1}$. This covariance matrix provides the statistics of the final position estimates in terms of an error ellipse with semimajor axis a and semiminor axis b. These are described in [10] as

$$Q_0 = \begin{bmatrix} \sigma_x^2 & \rho_{xy} \\ \rho_{xy} & \sigma_y^2 \end{bmatrix}$$

and

$$a^2 = \frac{2(\sigma_x^2 \sigma_y^2 - \rho_{xy}^2)}{\sigma_x^2 + \sigma_y^2 - [(\sigma_x^2 - \sigma_y^2)^2 + 4\rho_{xy}^2)]^{1/2}} \tag{8.4}$$

$$b^2 = \frac{2(\sigma_x^2 \sigma_y^2 - \rho_{xy}^2)}{\sigma_x^2 + \sigma_y^2 + [(\sigma_x^2 - \sigma_y^2)^2 + 4\rho_{xy}^2)]^{1/2}} \tag{8.5}$$

Subsequent sections in this chapter will use this basic form of atomic multilateration as a building block for more elaborate ad hoc localization schemes.

Finally, we note that besides least squares, other optimization methods can also be used to solve the same localization problem. One popular approach is to model the problem as a mass spring problem. An example of such an approach is described in [12]. In this problem, distance measurements between nodes are represented as springs having an associated energy component. The aim of this problem is to find a set of poses such that the sum of energies for each spring is minimized.

8.3.3 Localization Accuracy Metrics

The accuracy for of the computed mean square error location is typically evaluated by comparison with the corresponding Cramér–Rao bound (CR bound). The CR bound is a classical result from statistics that gives a lower bound on the error covariance matrix for an unbiased estimate of an estimated parameter. This is obtained by taking the inverse of the Fisher information matrix as defined in [33]. An example of how this bound is derived and used can be found in [23].

Other metrics of accuracy are the circular error probability (CEP) and geometric dilution of precision (GDOP). CEP is a function of the error covariance matrix of estimated locations that can be approximated by

$$CEP \cong (3/4)\sqrt{a^2 + b^2}$$

where a and b are the axes of the error ellipse described in Equations (8.4) and (8.5). GDOP measures the effect of the geometric configuration of the reference points on the location estimates and it is defined as the ratio of the rms error in position estimate and the rms distance measurement error:

$$GDOP = \frac{\sqrt{\sigma_x^2 + \sigma_y^2 + \sigma_z^2}}{\sigma_r}$$

GDOP is also related to CEP by $CEP \cong (0.75\sigma_r)GDOP$.

8.4 AD HOC TECHNIQUES FOR LOCATION DISCOVERY

The highly dynamic nature of mobile wireless devices calls for the development of localization techniques for ad hoc setups. In many situations, it is very hard to provide for infrastructure support that guarantees the availability of beacon nodes or other landmarks at all times. It is therefore very desirable to develop ad hoc location discovery techniques that will operate in the same set of dynamic conditions as ad hoc communication protocols. Some setups that motivate the development of ad hoc localization techniques include:

- **Randomly deployed nodes.** In the sensor network domain, it is often the case that GPS does not work in all places or its use may be prohibitive due to cost and power requirements. Under these constraints, ad hoc localization techniques are more suitable for determining the position of sensor nodes.
- **Rapid infrastructure installation.** In infrastructure settings, there is usually considerable cost and delay associated with the installation and calibration of a localization system. By using ad-hoc location discovery techniques, the system could self-calibrate, thus reducing cost and delay overheads.
- **Localization in the presence of obstacles in highly dynamic environments.** infrastructure environments, ad hoc techniques can also play an important role by assisting localization in the presence of obstacles in highly dynamic systems. For example, the ultrasonic signals used in many indoor localization systems cannot penetrate matter and, thus, cannot make measurements if they do not have direct line of sight with a set of beacons. Ad hoc location discovery techniques can alleviate this problem by allowing the localization of objects using indirect line of sight measurement information.

8.4.1 Challenges in Ad Hoc Localization Systems

Despite its several advantages, ad hoc location discovery also imposes a new set of challenges that need to be addressed. First, from an algorithmic perspective, the solution is often required to be computationally lightweight so that it can be performed on resource-constrained embedded microprocessors. At the same time, the solution should operate in a fully distributed manner that is tolerant of node failures. Furthermore, in situations where known landmarks are not within range of the nodes, individual nodes have to collaborate over multiple hops and utilize indirect line of sight measurement information to estimate their positions. This intensifies the problem, especially when network density and the density of landmarks becomes very sparse. Second, the ad hoc deployment of nodes in unknown environments affects the measurements at the physical layer. Transmissions from multiple sources can cause interference and changes in the surrounding terrain can introduce unexpected multipath and shadowing components, whereas changes in wheather conditions can affect signal propagation properties. Furthermore, the non-line-of-sight components increase in environments with more obstacles, introducing additional measurement errors. Third, despite the fact that many sensors and measurement technologies are readily available, a significant system integration effort is still required. To achieve the localization task, nodes need to coordinate and several components at multiple levels need to operate in harmony to produce an operational system.

8.4.2 Existing Ad Hoc Localization Approaches

Because of the aforementioned challenges and the great diversity of application requirements, there is currently no universal ad hoc localization system that will satisfy all applications. Instead, different types of systems have been proposed, each focusing on the requirements of the application at hand. Some setups require that positions be calculated at a central processing center in the network, and other setups call for fully distributed operation. The context in which locations are presented is also important. For some applications, relative positioning of nodes with respect to each other is sufficient, but for other applications a global frame of reference is more suitable. A local frame of reference typically does not require any prior knowledge of position information. When a global frame of reference is needed, the positions of some of the nodes, frequently referred as *anchors* or *beacons,* should be known. The existence of beacon nodes is typically assumed in most of the multihop localization approaches aiming to localize a number of nodes by using a very small fraction of beacons. Another difference in the existing approaches deals with the type of measurements. Some approaches such as the ones in [3] and [6] are based on connectivity only, whereas others are based on crude signal-strength measurements or very accurate ToF measurements.

The following subsections survey seven approaches that represent the recently proposed work in ad hoc localization algorithms and systems. The first two approaches are based on mere radio connectivity to provide node proximity. The next two approaches deal with systems that construct a local coordinate system without the use of beacons. The remaining three approaches deal with systems in which a small percentage of nodes are aware of their locations and act as beacons for the remaining nodes and positions of the nodes.

8.4.2.1 GPS-less Low-Cost Outdoor Localization System for Very Small Devices (GPSLC). This system determines the proximity of a node based on a set of predeployed location-aware reference nodes that transmit spatially overlapping beacon signals. Nodes localize themselves at the centroids of the reference nodes, from which they can receive beacon signals. The accuracy of localization depends on the density of the reference nodes and their transmission range. The best results are obtained when beacon nodes are arranged in a mesh pattern.

8.4.2.2 Convex Position Estimation in Wireless Sensor Networks (CPE).

The convex position estimation algorithm described in [6] computes the locations of nodes in an ad hoc network by performing computation at a central point in the network. Location estimation is formulated as a linear program (LP) or a semidefinite program (SDP), and the solutions are computed using a special optimization software package. Using this formulation, node locations are estimated using mere radio connectivity. More accurate position estimates can also be obtained if internode distance measurements are also known.

The convex optimization protocols consider two main constraint models, a *radial constraint model* and an *angular constraint model*. The radial constraint model represents RF node connectivity. A node is defined to be within transmission range if it is found within a circle of radius *R* from the transmitting node. Combining the individual constraints results (see Figure 8.3) in a reduced feasible region where an unknown node can be found. The angular constraint model applies to sensor nodes that use optical communication. In this case, the beam angle of a laser transmitter is modeled as a cone (or a triangle in two dimensions), with a certain beam angle θ and a finite length representing the maximum communication range.

According to the simulation results, when nodes with known locations are carefully placed on the perimeter of the network, position accuracies between 0.72 and 0.64 R are possible. This result is obtained when each node in the network has an average of 5.6 neighbors with 10% of the nodes acting as beacons. The authors have also shown that accuracies of less than 0.1 R are also possible at higher node densities and higher percentages of beacon nodes. One potential drawback of this approach is that the LP and SDP solutions require rigorous computation and can only be performed at a central point in the network.

8.4.2.3 GPS-Free Positioning in Mobile Ad Hoc Networks (GPSFP).

The system described in [4] and developed as part of the Terminode project [11] uses radio ToF measurements to provide locations in mobile ad hoc networks. Despite the existence of measurement errors, this system is reported to support mobile nodes with speeds up to 20 m/s and can provide adequate location accuracies for supporting basic network services such as location-aided routing.

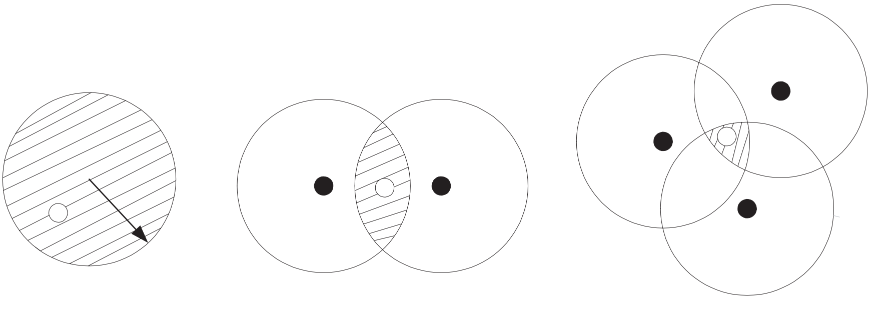


Figure 8.3. Combining radial constraints.

The Self-Positioning Algorithm (SPA) described in this paper forms local coordinate systems for each node and then merges them to construct a global coordinate system. In this setup, nodes initially discover their neighbors using a set of beacon signals. Each node then measures the distances to its one-hop neighbors using ToA and broadcasts these to all its neighbors. The nodes use this information to derive their local coordinate system. This is illustrated in Figure 8.4. First, node i constructs its own local coordinate system with nodes p and q. The corresponding coordinates are

$$i_x = 0; \qquad i_y = 0$$

$$p_x = d_{ip}; \qquad p_y = 0$$

$$q_x = d_{iq} \cos \gamma; \qquad q_y = d_{iq} \sin \gamma$$

where γ is the angle $\angle(p, i, q)$ and is obtained from the cosines rule:

$$\gamma = \arccos \frac{d_{iq}^2 + d_{ip}^2 - d_{pq}^2}{2d_{iq}d_{ip}}$$

The positions of other nodes with known distances to nodes i, p, and q can also be found using a similar set of rules. The position for node j in the example figure is given by

$$j_x = d_{ij} \cos \alpha_j \tag{8.6}$$

$$\text{if } \beta_j = |\alpha_j - \gamma| => d_{ij} \sin \alpha_j \tag{8.7}$$

$$\text{else} => j_y = -d_{ij} \sin \alpha_j \tag{8.8}$$

After the local coordinate systems are formed, applying a rotation or a rotation followed by a mirror transformation aligns the directions of all the local coordinate systems so that the

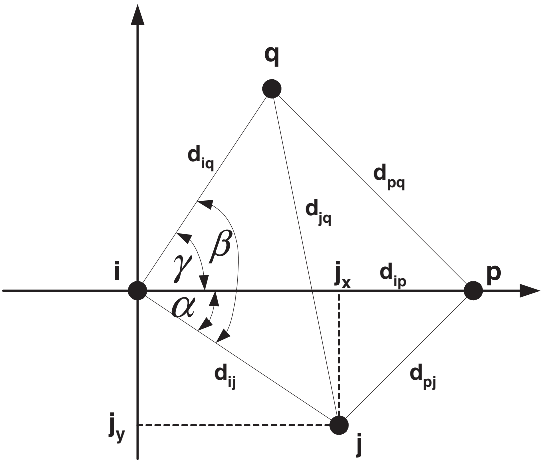


Figure 8.4. Local coordinate system formation example.

x and *y* coordinates point in the same direction for all the nodes. These transformations are performed among nodes that are one-hop neighbors, referred to as the local view set (LVS).

Once all the coordinate systems have the same direction, the nodes adjust their coordinates of a single node *i* that becomes the center of the coordinate system. This adjustment is initially done between nodes that are within the two-hop neighborhood from the node that becomes the origin of the coordinate system. This is illustrated in Figure 8.5. Node *i* knows the position of node *k* in terms of its own coordinate system, and node *k* knows the position of node *1* in its own coordinate system. The coordinates of node *1* can therefore be given in term of the coordinate system of node *i* by summing the corresponding vectors. When the coordinate systems of the two-hop neighborhood nodes are transformed, the same operation is applied to nodes that are further away, until all the nodes are in the same coordinate system.

The authors also note that this global coordinate system is not very suitable for networks with high levels of mobility. To address this problem, an alternative approach that computes the center of the coordinate system as a function of the node positions in the network is proposed. Instead of using a single node as the center of the coordinate system, a *location reference group* consisting of the nodes with the highest density in the network is selected, and the remaining nodes adjust their coordinate systems with respect to this group of nodes.

8.4.2.4 Locating Tiny Sensors In Time and Space: A Case Study (LTSTS).

This subsection presents the implementation of an operational localization system that works in close coordination with a time synchronization service. This design is validated in a testbed setup consisting of two types of nodes forming a tiered network. The first type of node is small, cheap, and computationally limited, and the second type of node consists of larger, faster, and more expensive nodes that act as "bases" for the smaller nodes.

The localization subsystem consists of three main components: a *wideband acoustic ranging system,* a *local coordinate system* algorithm, and *a location service.*

The ranging system (described in detail in [7]) uses a fine-grained time synchronization service and a wideband pseudonoise sequence to measure ToF of an acoustic signal

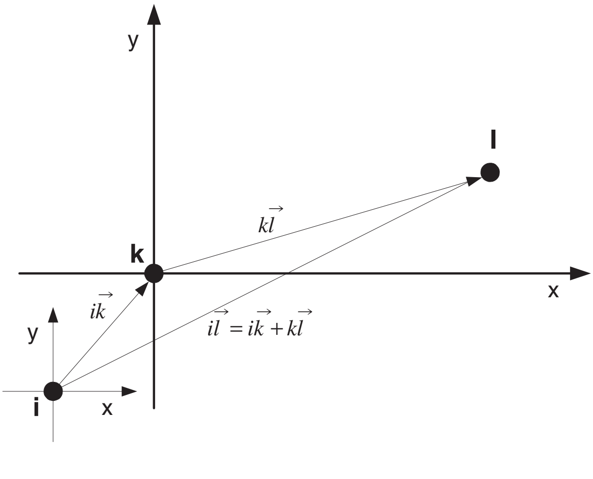


Figure 8.5. Local coordinate system formation example.

between a pair of nodes. The wavelengths of the emitted acoustic signals span from one centimeter to one meter, offering high resilience to signal scattering. Furthermore, the selection of orthogonal codes among different emitters allows the signals to be detected at the receiver even when collisions take place. This design choice simplifies the associated implementation complexity because it eliminates the need for tight coordination and synchronization between senders and receivers. The transmitter initiates the distance measurement by advertising to other devices its intention to transmit an acoustic signal with a particular code at a given time. The receiver starts sampling the acoustic channel at the specified time and the sampled time series is compared to a locally generated signal using a sliding correlator. A portion of the observed signal aligned with the reference signal at the "best" offset is shown in Figure 8.6.

Once the distance measurements are completed, a local coordinate system is established. Distance measurements are collected at a single aggregation point that uses a mass spring model to establish an initial coordinate system. Four fully connected, noncoplanar points close to the aggregation point are initially considered. The mass spring system provides an initial configuration for these nodes, which is then improved using nonlinear regression to minimize Gaussian measurement error. The RMS error for this set of nodes based on the testbed measurements is 11.5 cm. After the local coordinate system is formed, less powerful nodes can discover their location using the location service provided by the more powerful nodes. According to the testbed results, the RMS position error for these nodes is 9.2 cm.

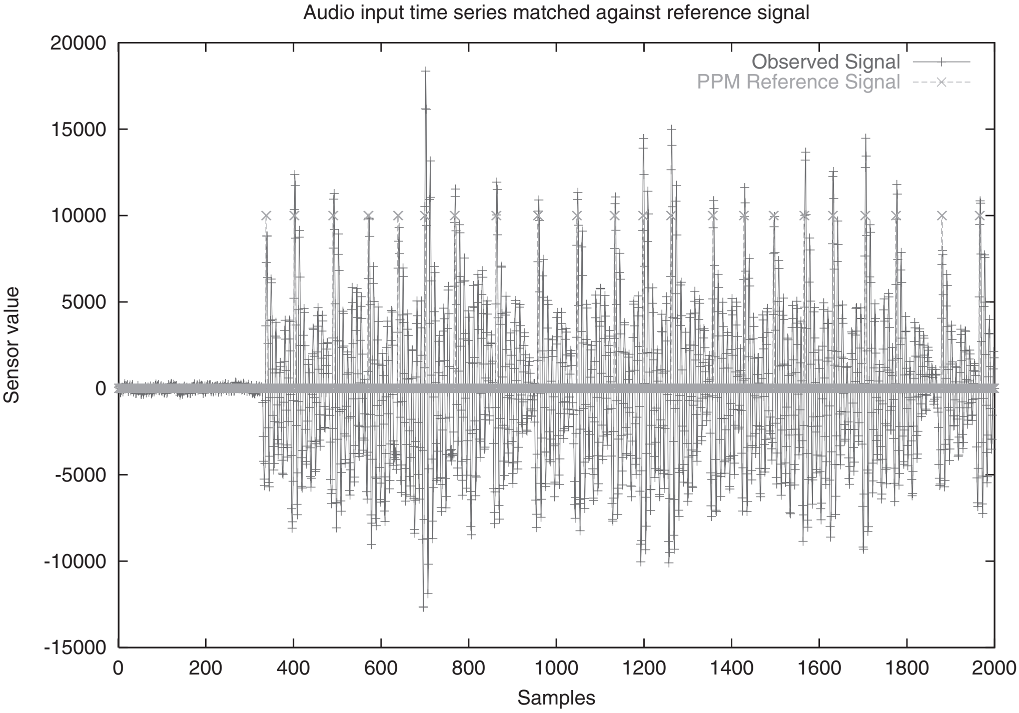


Figure 8.6. Pulse position modulated reference signal aligned with observed signal, captured under very low noise conditions.

8.4.2.5 A Self-Localization Method for Wireless Sensor Networks (SLM). Another notable localization method has been developed in [23]. In this work, Moses et al. have shown that sensor node positions and orientations can be estimated using signals from acoustic sources with unknown locations. Each acoustic source generates a known acoustic signal that is detected by the sensor nodes. The sensor nodes in turn measure the ToA and DoA of the signal and propagate this information to a central information-processing center (CIP). The CIP fuses the information using maximum likelihood estimation to obtain the location and orientation of the sensor nodes.

In addition to this algorithm, the authors of [23] also treat the case in which only partial measurements are available. This occurs when the source signal is detected only by a subset of the nodes or when some sensors have only one successful measurement of either ToA or DoA.

8.4.2.6 Ad Hoc Positioning System (APS). The Ad Hoc Positioning System proposed by Nicolescu and Nath in [21] estimates the locations in an ad hoc network by considering distances to a set of landmarks. This study explores three alternative propagation methods: *DV-hop, DV-distance,* and *Euclidean*.

In the DV-hop method, landmarks propagate their location information inside the network. Each node forwards the landmark information to its neighbors and maintains a table with the landmark identification (ID), location, and hop distance. When a landmark receives one of the propagated packets with the position of a different landmark, it uses that information to calculate the average hop distance between the two landmarks. The computed average hop distance is broadcast back into the network as a correction to previously known hop distances. The nodes that receive this message use the average hop distances to each of the landmarks to estimate their distances to the landmarks. This information is then used to triangulate the node location. The corrections are propagated in the network using controlled flooding. Each node will forward a correction from a certain landmark only once in an effort to ensure that nodes will receive only one correction from the closest landmark. This policy tries to account for anisotropies in the network.

The DV-distance approach is similar to DV-hop but uses radio received signal strength measurements to measure distances. Although this approach gives finer-level granularity, it is also the most sensitive to measurement error since the received signal strength is greatly influenced by the surrounding environment and is, therefore, not always consistent.

The Euclidean propagation method uses the true distance measurement to a landmark. In this case, nodes that have at least two distance measurements to nodes that have distance estimates to a landmark can use simple trigonometric relationships to estimate their locations.

The reported simulation results indicate that the DV-hop propagation method is the most accurate of the three and determines the positions of nodes within one-third of the radio range in dense networks.

8.4.2.7 Robust Positioning Algorithms for Distributed Ad Hoc Wireless Sensor Networks (RPAD). The algorithm described in [26] explores the same problem setup as APS. It estimates the positions of nodes in an ad hoc network by utilizing internode distance measurements and a set of *anchor* nodes. Position estimation is carried out in two phases: *startup* and *refinement*. The former provides a set of crude estimates for the node locations, whereas the latter refines these estimates using a least squares algorithm to obtain the final estimates.

The Hop-TERRAIN algorithm is similar to the DV-hop algorithm used in the APS system. First, each node in the network finds out the number of hops to each anchor node. When an anchor node receives the number of hops to another anchor, it then computes the average hop distance and broadcasts it back into the network. Nodes with unknown locations multiply the number of hops with the average hop distance to get an estimate for the distance to the anchor nodes. The nodes then perform a triangulation to obtain an initial estimate with respect to the anchor nodes. One potential problem with Hop-TERRAIN is that nodes with the same hop distances to the anchor nodes arrive at the same initial position estimate. This problem can be avoided by excluding one of the anchor points so that the nodes have at least one different reference point.

The refinement phase uses the estimated distances of the one-hop neighbors of each node to obtain more accurate position estimates. Refinement is an iterative process that uses a least squares solution. At each iteration, each node with unknown location computes and broadcasts its position estimate to its neighbors. The neighbor nodes use this information to update their estimate and rebroadcast it back to their neighbors where the new information is included in the computation of the next iteration. The algorithm terminates when a maximum number of iterations is completed.

The study of this refinement algorithm revealed two main problems:

1. Error propagation through the network.
2. Hard network topologies. These are cases there the whole or parts of the network can be rotated while keeping the intranode ranges. This problem was originally identified in [27] and will also be described in the next subsection.

To address the first issue, the authors perform the least squares computation using weights (see Section 8.3.2) that correspond to the confidence level for each location. The confidence levels vary between 0 and 1. Initially, anchor nodes have confidence level 1 and nodes with unknown locations have a confidence level 0.01. At each iteration of the algorithm, the confidence level of the node that computes a new estimate is set to the average of the confidence levels of its neighbors. In most cases, this gradually raises the confidence level of the nodes. If the triangulation fails due to insufficient constraints or because the computed result does not meet some other consistency checks, then the confidence level of the node is set to 0 so that other nodes in the next iteration do not use it. The node failing to improve its confidence level may try to recompute its location in subsequent iterations. If the computed result is rejected multiple times, then the node is excluded from the refinement process. The simulation results note that although the use of confidence level yields an improvement of the average error in computed locations, it cannot be used as an indicator of the estimated accuracy.

For the second problem of hard topologies, the authors propose a heuristic that detects most of the ill-connected configurations. In such configurations, although each unknown node has at least three neighbors, the whole network or parts of the network can be rotated while still maintaining a consistent pattern. This is illustrated in the example network topology of Figure 8.7, which originally appeared in [27]. Node 4 in the example has two possible positions that are consistent with the distance measurements. To determine if a certain topology is sound, the Hop-TERRAIN records the ID of each node's immediate neighbor on the shortest path to each anchor point. If multiple shortest paths are available, then the ID for the node on the first shortest path found is recorded. When the number of

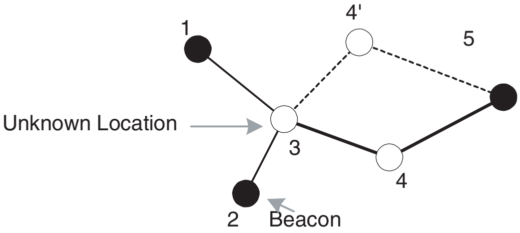


Figure 8.7. An example of an ill-constrained topology.

unique IDs in this set reaches three (four for three dimensional scenarios), the node declares itself sound and enters the refinement phase. The neighbors of the sound nodes add this node to their sound set and the process continues around the network. This process is repeated throughout the network and identifies the majority of ill-connected configurations.

According to the simulation results, the *robust positioning* algorithm achieves, on average, position errors of less than 33% of a node's radio range in the presence of a 5% range measurement error when at least 5% of the nodes are anchor nodes.

8.4.2.8 Ad Hoc Localization System (AHLoS). The Ad Hoc Localization System developed at the University of California, Los Angeles [27, 28] focuses on the fine-grained localization of sensor nodes in an ad hoc network. The core component of the system is collaborative multilateration, a method that allows nodes with unknown locations to collaborate with each other and jointly estimate their location based on some initial beacon information and a set of distance measurements between the nodes. With this method, the direct line of sight to beacons requirement is relaxed to an indirect line of sight to beacons requirement that enables nodes to estimate their locations even when beacon nodes are found multiple hops away. Collaborative multilateration is provided in two computation models, centralized and distributed. The centralized model operates based on a global view of the network. It uses all beacon node location information and all the internode distance measurements as constraints to set up a global nonlinear optimization problem that is then solved using iterative least squares. By considering all the constraints in the network at the same time, this solution minimizes error propagation incurred by the use of measurements over multiple hops.

The distributed computation model is an approximation of the centralized model that is also based on iterative least squares but distributes computation evenly to all nodes with unknown locations. This results in significantly less computation and provides a more robust setup for ad hoc networks. Furthermore, it enables small resource-constrained nodes to collaborate among each other to set up and solve a nonlinear optimization problem with local computation but with respect to the global constraints, a task that none of the nodes can perform individually.

Collaborative multilateration is based on a three-phase process. First, before position estimation can start, nodes need to organize themselves into groups. These groups, referred to as *collaborative subtrees,* ensure that all the unknown nodes included in each group are well constrained or overconstrained so that the estimation problem has a unique solution. The second phase uses simple geometric relations to estimate a set of *initial po-*

sition estimates for each unknown node. These estimates are used to initialize the third phase, a *refinement phase* that uses an iterative least squares algorithm. A good set of initial estimates is required to ensure that the iterative process does not get stuck at local minima.

Computing Initial Estimates. A set of initial estimates is obtained by applying the distance measurements as constraints on the x and y coordinates of the nodes. Figure 8.8a shows how the distance measurements from two beacons A and B can be used to obtain the x coordinate bounds for the unknown node C. If the distance between an unknown and the beacon A is α, the x coordinates of node C are bounded to the left and to the right of the x coordinate of the coordinate of beacon A, $x_A - \alpha$ and $x_A + \alpha$. Beacon B imposes its constraints in the same way. Similarly, if beacon B is two hops away from C (as in Figure 8.8b), the coordinates bounds of node C are defined by the length of the minimum weight path to C, $b + c$, so the bounds for C x-coordinates with respect to B are $x_B - (b + c)$ and $x_B + (b + c)$. By knowing this information, C can determine that its x coordinate bounds with respect to beacons A and B are $x_B + (b + c)$ and $x_A - \alpha$. This operation selects the tightest left-hand-side bound and the tightest right-hand-side bound from each beacon. The same operation is applied on the coordinates. The node then combines its bounds on the x and y coordinates to construct a bounding box of the re-

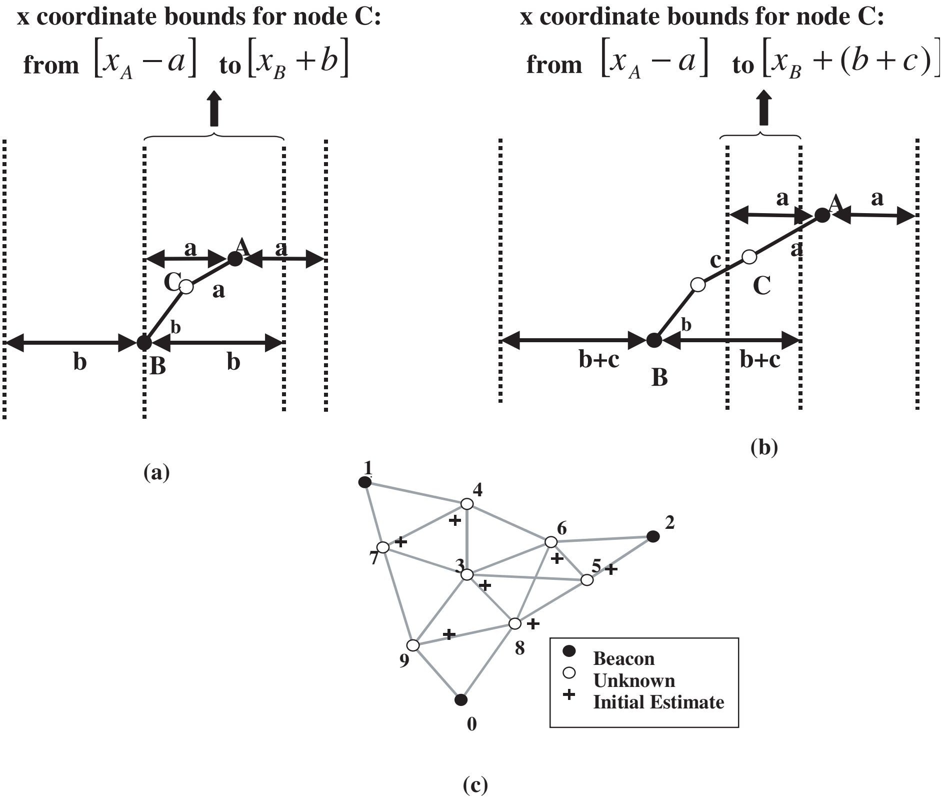

x coordinate bounds for node C:
from $[x_A - a]$ to $[x_B + b]$
a
A
a
C
a
b
b
B
b
(a)
x coordinate bounds for node C:
from $[x_A - a]$ to $[x_B + (b + c)]$
a
A
a
c
a
C
b
b+c
B
b+c
(b)
1
4
6
2
7
3
5
8
9
0
Beacon
Unknown
Initial Estimate
(c)

Figure 8.8. Obtaining initial position estimates.

gion where the node lies. To obtain this bounding box, the locations of all the beacons are forwarded to all unknowns along a minimum weight path. This forwarding is similar to distance vector routing (e.g., DSDV) but uses the measured distances instead of hops as weights.

The initial position estimate of a node is taken to be as the center of the bounding box. When these constraints are combined with the conditions for position uniqueness, they provide a good set of initial estimates for the iterative least squares algorithm used in the next phase. The resulting initial estimates for a 10-node network with three beacons is shown in Figure 8.8c. The initial estimates for each node (shown as crosses in the figure) are close to the actual node positions and provide a good starting point for the refinement process. One challenge in this method is that the quality of the initial estimates suffers when the unknown nodes lie far outside the convex hull formed by the beacons on the perimeter of the collaborative subtree, and then the initial estimates degrade. In our experiments with large networks, we deploy some of the beacons on the edges of the sensor field.

Centralized Computation Model. The centralized computation solved a nonlinear optimization problem by considering the global topology. This is illustrated by the example network in Figure 8.9. In this example node 1, 2, 5, and 6 are beacons and nodes 3 and 4 have unknown locations.

The nodes can perform a total of five distance measurements that provide the following five constraints:

$$f_{2,3} = r_{2,3} - \sqrt{(x_2 - ex_3)^2 + (y_2 - ry_3)^2}$$

$$f_{3,5} = r_{3,5} - \sqrt{(ex_3 - x_5)^2 + (ey_3 - y_5)^2}$$

$$f_{4,3} = r_{4,3} - \sqrt{(ex_4 - ex_3)^2 + (ey_4 - ey_3)^2}$$

$$f_{4,5} = r_{4,5} - \sqrt{(ex_4 - x_5)^2 + (ey_4 - y_5)^2}$$

$$f_{4,1} = r_{4,1} - \sqrt{(ex_4 - x_1)^2 + (ey_4 - y_1)^2}$$

Solving a similar objective function as in the case of the single-hop setup described in Section 7.3.1 we have

$$F(x_3, y_3, x_4, y_4) = \min \sum f_{i,j}^2$$

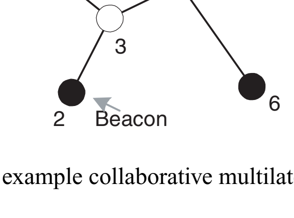


Figure 8.9. An example collaborative multilateration topology.

The difference from the single-hop setup is that that two types of measurements need to be considered, beacon-unknown and unknown-unknown. The former measurement is the same type as the one described by Equation 8.3. The latter involves two unknowns that use each other as an anchor point, and the Taylor expansion of this has the form

$$r_{i,k} = f_i^0 + \Delta x_a \delta x_b + \Delta y_a \delta y_b + \Delta x_b \delta x_a + \Delta y_b \, \delta y_a + O(\Delta^2) \tag{8.9}$$

Note that both nodes *a* and *b* have unknown locations. The position estimates can still be computed using an iterative solution similar to the one described in Section 8.3.4. The matrices for the setup in Figure 8.9 will be

$$\delta = \begin{bmatrix} \delta x_a \\ \delta y_a \\ \delta x_b \\ \delta y_b \end{bmatrix}, \qquad A = \begin{bmatrix} \Delta x_{1,a} & \Delta y_{1,a} & 0 & 0 \\ \Delta x_{2,a} & \Delta y_{2,a} & 0 & 0 \\ 0 & 0 & \Delta x_{3,b} & \Delta y_{3,b} \\ 0 & 0 & \Delta x_{4,b} & \Delta y_{4,b} \\ \Delta x_{b,a} & \Delta y_{b,a} & \Delta x_{a,b} & \Delta y_{a,b} \end{bmatrix} \qquad \text{and} \qquad z = \begin{bmatrix} r_{1,a} - f_{1,a}^{(0)} \\ r_{2,a} - f_{2,a}^{(0)} \\ r_{3,b} - f_{3,b}^{(0)} \\ r_{4,b} - f_{4,b}^{(0)} \\ r_{a,b} - f_{a,b}^{(0)} \end{bmatrix}$$

Distributed Computation. The distributed computation model is an approximation to the centralized computation model described in the previous section. Instead of estimating locations by considering all measurements at once, each node is responsible for computing its own position estimate based on the current estimate of its neighboring nodes. In this algorithm, each node in the network uses its neighboring nodes (both beacons and nodes with unknown locations) to estimate its location. First the node generates a new estimate of its location using atomic multilateration based on the current position estimates of its neighbors. Once the computation is completed, the node then forwards its new position estimate to its neighbors. The neighbors with unknown locations use this information to generate a new estimate of their locations. By repeating this action across a well-constrained configuration of nodes, the nodes can estimate their locations over multiple iterations.

If the well-constrained configuration has more than two nodes with unknown locations, then all the nodes should generate their position updates at the same rate so that a gradient with respect to the global constraints is formed. This can be achieved by having nodes generate their updates sequentially using a depth-first traversal. The algorithm repeats until all nodes with unknown locations reache a prespecified tolerance.

Computation and Communication Tradeoffs. Besides robustness, the distributed computation model results in considerable savings in computation. This is because in the distributed model the matrices grow with respect to the number of neighbors of a node. In the centralized computation model, the matrices grow much faster since their size depends on the number of unknowns and the number of nodes in the network. Figure 8.10 shows a comparison of the total number of floating point operations required by MATLAB to solve the same network using the centralized and distributed computation models. The distributed computation model provides a more scalable behavior in the number of nodes, which is highly desirable in an ad hoc setup.

In addition to the savings in computation, the distributed approach also has favorable results on the communication patterns. Since nodes only have to communicate occasion-

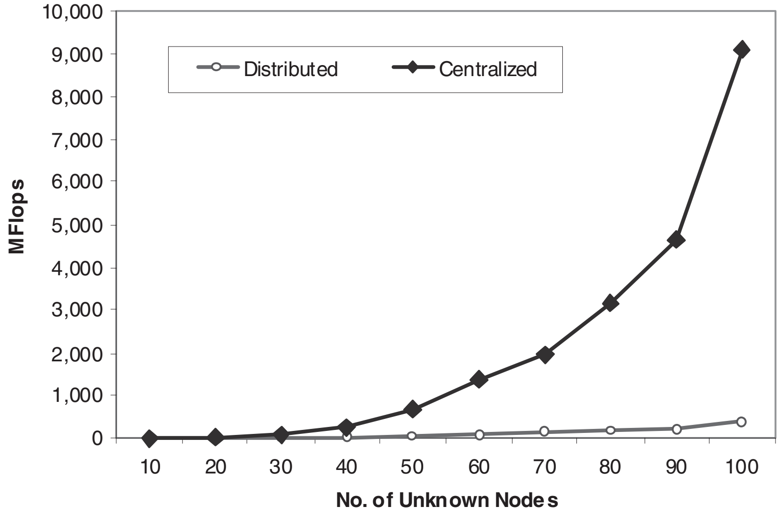


Figure 8.10. Computation cost comparison between centralized and distributed models.

ally with their one-hop neighbors, the communication overhead is evenly divided among all nodes. From an energy perspective, this is desirable since it avoids the uneven power consumption caused by the forwarding of packets across a multihop network.

8.5 FUTURE DIRECTIONS IN LOCATION DISCOVERY

With the new technological developments, the application space of location-based applications is exploding. Despite this progress and the recent work described in this chapter, localization in the ad hoc setup still imposes a set of challenges at many different levels. At the physical layer, controlling measurement error in different environments is an issue. Many researchers are considering the fusion of measurements from orthogonal sensing modalities in an effort to reduce measurement uncertainties. Others are focusing on the development of new measurement methods, and other research studies the error behavior characteristics and provides network-level algorithms location estimates [29]. Sensor transducer calibration is another important issue that researchers try to handle at the network level. Recent efforts in this direction are described in [32].

In addition to the position estimation challenges, localization systems and protocols need to be tightly integrated with other protocols and applications. These protocols should handle transitions between different technologies and different location accuracies as nodes move into environments (or hierarchical systems) supporting a diverse set of location discovery mechanisms.

With such mechanisms in place, the application space of location-based services is expected to advance very rapidly, enabling a new era of context-aware, ubiquitous computation and communication (see Table 8.1). Such knowledge of fine-grained location information about people and devices, however, will also impose additional problems

Table 8.1. Ad Hoc Localization System Summary

System	Measurement technology	Main characteristic	Reported accuracy
GPSLC	RF connectivity	Proximity, Distributed operation	Depends on beacon pattern
CPE	RF connectivity	Proximity, Requires rigorous centralized computation	0.64–0.72 tx range
GPSFP	RF ToF	Distributed operation, local coordinate system	N/A
LTSTS	Acoustic ToF	Centralized opeation, constructs a local coordinate system	11.5 RMS error for bases, 9.3 RMS error for small nodes
SLM&	Acoustic ToF	Centralized operation	0.35 m
APS	Radio signal strength	Distributed operation	1/3 tx range
RPAD	Radio signal strength or acoustic	Distributed operation	1/3 randio range at 5% beacons
AHLoS	Ultrasound ToF	Distributed operation	3–5 cm

regarding security and privacy. These issues should be carefully considered before many types of localization systems are deployed on a large scale.

REFERENCES

1. The Active Badge System, http://www.uk.research.att.com/ab.html.
2. P. Bahl, V. Padmanabhan, "An In-Building RF-based User Location and Tracking System," *Proceedings of INFOCOM 2000,* Tel Aviv, Israel, vol 2, pp. 775–784, March 2000.
3. N. Bulusu, J. Heidemann, and D. Estrin, "GPS-Less Low Cost Outdoor Localization For Very Small Devices," *IEEE Personal Communications Magazine,* Special Issue on Networking the Physical World, August 2000.
4. S. Capkun, M. Hamdi, and J. P. Hubaux, "GPS-Free Positioning in Mobile Ad-Hoc Networks," in *Proceedings of Hawaii International Conference on System Sciences,* HICCSS-34 Jan, 2001.
5. P. Castro, P. Chiu, and R. Muntz, "A Probabilistic Location Service for Wireless Network Environments," in *Proceedings of Ubicomp 2001.*
6. L. Doherty, L. El Ghaoui, and K. S. J. Pister, "Convex Position Estimation in Wireless Sensor Networks," in *Proceedings of INFOCOM 2001,* Anchorage, AK, April 2001.
7. L. Girod and D. Estrin, "Robust Range Estimation Using Acoustic Multimodal Sensing," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS),* Maui, Hawaii, October 2001.
8. R. Fontana, A. Ameti, E. Richley, L. Beard, and D. Guy, "Recent Advances In Ultra Wideband Communications Systems," in *Proceedings of IEEE Conference on Ultra Wideband Systems and Technologies,* Baltimore, May 2002.
9. R. Fontana and S. Gunderson, "Ultra Wideband Precision Asset Location System," in *Proceedings of IEEE Conference on Ultra Wideband Systems and Technologies,* May 2002.

10. W. Foy, “Position-Location Solution by Taylor Series Estimation,” *IEEE Transactions on Aerospace and Electronic Systems, AES-12,* 2, 187–193, March 1976.

11. J.-P. Hubaux, J. -Y. Le Boudec, S. Giordano, M. Hamdi, Lj. Blazevic, L. Buttyan, and M. Vojnovic, “Towards Mobile Ad-Hoc WANs: Terminodes,” *IEEE WCNC,* September 2000.

12. A. Howard, M. Mataric, and G. Sukhatme, “Relaxation on a Mesh: A Formalism for Generalized Localization,” in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS),* Wailea, Hawaii, Oct. 2001.

13. J. Hightower and G. Boriello, “Location Systems for Ubiquitous Computing,” *IEEE Computer, 34*(8), 57–66, Aug 2001.

14. T. Imielinski and J. Navas, “GPS-Based Geographic Addressing, Routing and Resource Discovery,” *Communications of the ACM, 42,* 4, April 1999.

15. Intersense, Inc., http://www.isense.com.

16. E. Kaplan, *Understanding GPS Principles and Applications,* Artech House, 1996.

17. B. Karp and H. T. Kung, “GPSR: Greedy Perimeter Stateless Routing for Wireless Networks,” in *Proceedings of Mobicom 2000.*

18. A. Ladd, K. Bekris, A. Rudys, G. Marceau, L. E. Kavraki, and S. Dan, “Robotics-Based Location Sensing Using Wireless Ethernet,” in *Proceedings of Mobicom,* Atlanta, Georgia, September 2002.

19. J. Li, J. Jannotti, D. S. J. De Couto, D. Karger, and R. Morris, “A Scalable Location Service for Geographic Ad-Hoc Routing,” in *Proceedings of Mobicom 2000.*

20. A. Nasipuri and K. Li, “A Directionality Based Location Scheme for Wireless Sensor Networks,” in *Proceedings of First ACM International Workshop on Wireless Sensor Networks and Applications,* pp. 105–111, September 28, Atlanta, Georgia 2002.

21. D. Nicolescu and B. Nath, “Ad-Hoc Positioning System,” in *Proceedings of IEEE GlobeCom,* November 2001.

22. M. Mauve, J. Widmer, and H, Hartenstein, “A Survey on Position Based Routing in Mobile Adhoc Networks,” *IEEE Network Magazine, 15*(6), 30–39, November 2001.

23. R. L. Moses and R. M. Patterson, “Self-calibration of sensor networks,” in *Unattended Ground Sensors Technologies and Applications IV* (Proceedings of SPIE, Vol. 4743), E. M. Carapezza (Ed.), pp. 108–119, April 1–4, 2002.

24. N. Priyantha, A Chakraborthy, and H. Balakrishnan, “The Cricket Location Support System,” in *Proceedings of International Conference on Mobile Computing and Networking,* pp. 32–43, Boston, August 2000.

25. RF Technologies, http://www.rftechnologies.com/pinpoint.

26. C. Savarese, J. Rabay, and K. Langendoen, “Robust Positioning Algorithms for Distributed Ad-Hoc Wireless Sensor Networks,” in *Proceedigs of USENIX Technical Annual Conference,* June 2002.

27. A. Savvides, C. C. Han, and M.B. Srivastava, “Dynamic Fine-Grained Localization in Ad-Hoc Networks of Sensors,” in *Proceedings of Fifth Annual International Conference on Mobile Computing and Networking, Mobicom,* pp. 166–179, Rome, Italy, July 2001.

28. A. Savvides, H. Park, and M. B. Srivastava, “The Bits and Flops of the n-Hop Multilateration Primitive for Node localization Problems,” in *Proceedings of the First International Conference on Wireless Sensor Networks and Applications, Mobicom,* September 2002.

29. S. Slijepcevic, S. Megerian, and M. Potkonjak. “Location Errors in Wireless Embedded Sensor Networks: Sources, Models, and Effects on Applications.” *ACM Mobile Computing and Communications Review, 6,* 3, 67–78, 2002.

30. R. Wand, A. Hopper, V. Falcao, and J. Gibbons, “The Active Bat Location System,” *ACM Transactions on Information Systems,* pp. 91–102, January 1992.

31. WhereNet, http://www.wherenet.com.

32. K. Whitehouse and D. Culler, "Calibration as Parameter Estimation in Sensor Networks," in *Proceedings of the First ACM International Workshop on Wireless Sensor Networks and Applications,* pp. 59–67, Atlanta, Georgia, September 2002.

33. H. L. Van Trees, Chapter 12 in *Detection Estimation and Modulation Theory, Part I,* Wiley, 1968.

34. VOR System http://www.navfltsm.addr.com/vor-nav.htm.

CHAPTER 9

MOBILE AD HOC NETWORKS (MANETs): ROUTING TECHNOLOGY FOR DYNAMIC WIRELESS NETWORKING

JOSEPH P. MACKER and M. SCOTT CORSON

The fact that mobile networking is a "hot" topic of present technology research and development hardly needs stating. Recent technical publications are inundated with reports of promising technologies and approaches for a better wireless future. This is understandable, as wireless networking is expanding into many varied dimensions of application space. Wireless data networking was once viewed as a highly limited business, with access obtainable only by privileged players or amateur operators. Yet, it is now seen as a rapidly expanding general business and public resource. Much of this is due to the proliferation of low-cost, low-power, high-capacity wireless local area network (WLAN) technologies. The reusability of the radio spectrum and widespread deployment of the unlicensed spectrum is now standard practice across society and is being applied within a diverse set of industries. With the recent explosion of inexpensive home and business wireless networking turnkey products, nontechnical network edge users are increasingly becoming wireless network owners and operators. And despite the fact that technical coexistence problems certainly arise from highly uncoordinated deployments of unlicensed wireless systems, the technology is here to stay.

In the not so distant past, wireless networking technologies relied exclusively on significant infrastructure planning and a strict hierarchy of control for typical operations. With the expected tremendous growth rate of wireless applications, especially in the hands of independent end users, businesses, and communities, it will be beneficial to have more adaptive, self-organizing technologies that robustly operate and adapt to changes (minor or severe) within a network region. Mobile ad hoc networking (MANET) is one area of evolving

Mobile Ad Hoc Networking. Edited by Basagni, Conti, Giordano, and Stojmenovic.
ISBN 0-471-37313-3 © 2004 Institute of Electrical and Electronics Engineers, Inc.

technology that can support the operation of more adaptive wireless networks. The overall aim of this chapter is first provide a brief introduction to MANET technology and then provide some of the authors' perspectives on several related issues as follows:

- Where and when does MANET technology make sense to consider?
- What are some present or envisioned applications for MANET?
- What related work has gone on within the standards community?
- What are some future issues and directions for MANET-related technology?

9.1 INTERNET LAYER ROUTING AND DYNAMIC WIRELESS NETWORKS

Computer and communication networks such as the Internet are multilayered, complex systems relying on many different protocols and associated algorithms for seamless, reliable operation. As networks extend beyond direct-link connection cases (e.g. a local area network in which all nodes are logical single-hop neighbors), there is a need for some function to forward traffic on the behalf of source systems to destination systems that are out of direct connectivity range. Within the Internet Protocol (IP) suite, IP *routing* technology is typically used to direct the forwarding of such traffic. Several of the technical fundamentals of routing involve how protocols *find, manage, and use* multihop paths for forwarding information on the behalf of specific end systems to particular destination systems.

The global internetwork (i.e., the Internet) routing system must adapt on some reasonable time scale to changes and failures in the network infrastructure, and it must scale to support many billions of end systems. Thus far, the Internet routing system has evolved to largely meet these expectations, yet the proliferation of inexpensive wireless technologies, portable computing, and the information-hungry nature of our increasingly mobile society pose new challenges and opportunities. New design challenges are evident at the *edges* of the Internet infrastructure, where wireless technologies are being rapidly deployed in the hands of users, organizations, and communities to expand local Internet connections, and provide network service for a wide variety of information devices and users. The Internet Protocol (IP) core design tenets—connectionless networking and packet-based forwarding—are ideally suited for use in highly dynamic contexts such as mobile wireless. Yet, new technology developments and design extensions that better address and meet the unique challenges and opportunities of wireless operation are needed.

MANET can enable improved dynamic wireless operation by addressing routing technology improvements within this context. We define MANET operational regions as collections of wireless network platforms or "nodes," where nodes may organize and maintain a routing infrastructure among themselves in a relatively arbitrary fashion. Due to the fundamental dynamic nature of wireless network communications, it is not necessary that nodes be in motion for this to be a valuable capability, but the general design assumption is that relative node mobility should be directly supportable. MANET nodes are enabled with the potential for wireless-compatible routing capabilities. With these technology enhancements, nodes can more effectively monitor and adapt to changes in the local neighborhoods and across MANET topological regions of operation. Figure 9.1 shows a simple example of the need for dynamic routing when a wireless topology change occurs. In this case, routing provides the functionality to forward traffic from node A to node F. As dynamics cause the achievable network topology to change (e.g., node movements, wireless

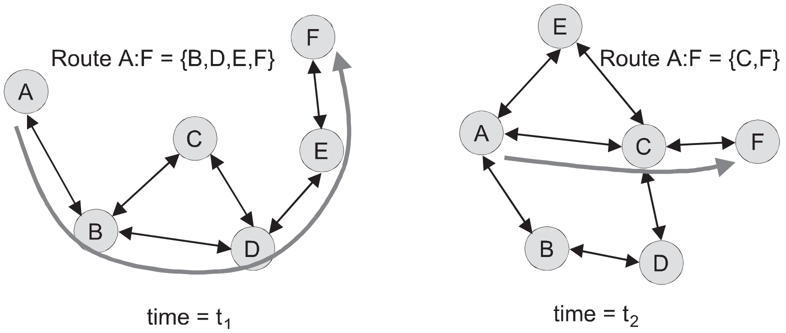


Figure 9.1. Dynamic routing in a changing topology.

link failures), valid routes must be discovered and maintained in order to forward network data to the desired destination, node F in this example. This capability is no different from the general goal of IP layer routing, but the underlying design assumption of wireless interfaces and possibly mobile routing nodes presents increased technical challenges.

Overall, MANET deployments have been envisioned in many different scenarios and on many different scales. In considering use of a particular MANET protocol or approach, it is important to be cognizant of operational parameters that can directly affect suitability and performance. Some operational parameters that affect overall performance and scalability include number of peer routing nodes, type and degree of link dynamics, expected user traffic patterns, network density, and lower-layer technology characteristics. The interplay of all these parameters and their relative performance effects can be quite complex and different for each routing approach under consideration. Aware of the oversimplification and somewhat arbitrary nature of the following terminology, we define some rough scalability regimes based on the number of peer routers within a region to aid discussion:

- Small-scale (i.e., 2–29 nodes)
- Moderate-scale (i.e., 30–100 nodes)
- Large-scale (i.e., 100+ nodes)
- Very large scale (i.e., 1000+ nodes)

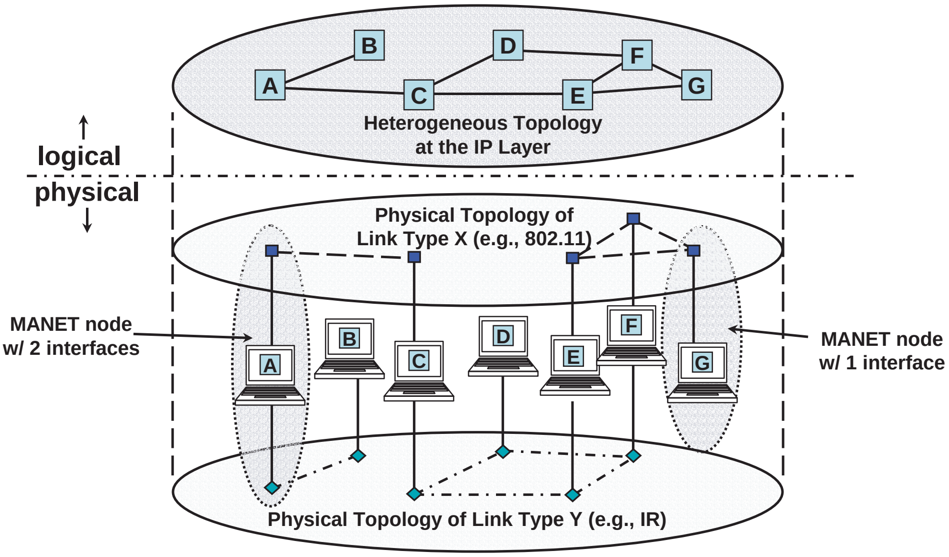
We are applying this terminology within a single operating region, and, therefore, a deployment of 10 moderate-scale MANET network operating regions consisting of 100 nodes each (1000 total nodes) is not a very large scale MANET in this terminology, but is a considered collection of moderate-scale networks. At the time of this writing, MANET operational experience has been gathered in small- to moderate-scale routing region cases. Early experimentation and use of various routing schemes has been performed on a variety of working hardware and software operating systems. In addition to working MANET systems, a large number of independent simulation models have been developed and numerous performance studies have been performed, mostly at the moderate-scale and some at the large-scale level across a wide variety of protocol types and within a range of available network simulation packages [9–11]. In the authors' opinion, there remains a growing amount of promising future technical work to be done in the area of creating *large to very large* MANET region technology, but even with some early implementations much of this scalability work remains at a research stage with many practical issues regarding performance to be further explored. On the other hand, small-to-moderate-scale mobile network applications are reaching a level of understanding and maturity to be operationally viable in a wide variety of scenarios.

Before we address further performance and application issues related to MANETs, let us discuss some of the motivation behind developing MANET technology specifically for the Internet protocol suite.

9.1.1 Why the Internet Protocol (IP) Layer?

The following fundamental question has been asked many times and deserves consideration here before we continue: "Why is it especially important to solve mobile routing problems at the IP layer?" It is clear that mobility-enhanced routing functionality can be developed at lower layers of a protocol stack (i.e., below the IP layer). Examples of such prior work include HIPERLAN 1 and lower-layer cluster-based routing [12, 13]. Such subnet convergence technology can provide a logical local area network (LAN) appearance to the IP layer and handle a degree of packet relaying. To the authors, it becomes not a simple question of where the technology belongs. Lower-layer approaches remain valid engineering design options, and improving routing functionality at different layers of a protocol stack can support different architectural needs. There remain several key reasons to consider and promote fundamental dynamic routing enhancements at the IP layer.

First, the main reason for doing work at the IP layer is to better support heterogeneity and networked interoperability of lower-layer technologies—*to continue to build a network of networks.* Multiple link layers may be simultaneously deployed, the physical composition of which can form a unified, logical whole via routing at the IP layer, as illustrated in Figure 9.2. The number of commercially available wireless technologies and devices



B
D
F
A
C
E
G
Heterogeneous Topology at the IP Layer
logical
physical
Physical Topology of Link Type X (e.g., 802.11)
MANET node w/ 2 interfaces
MANET node w/ 1 interface
A
B
C
D
E
F
G
Physical Topology of Link Type Y (e.g., IR)

Figure 9.2. A heterogeneous mix of MANET node and interface types.

continues to increase over time [e.g., infrared, 802.11, Bluetooth, ultrawideband (UWB)]. We expect lower-layer wireless technologies to continue to evolve and vary in use and popularity, yet IP tends to remain as the fundamental internetworking glue working across multiple technologies. As this understanding permeates the industry, we predict that link layers will be more expressly designed to plug into the underbelly of an IP network. Second, any IP layer software development often capitalizes on the rich variety of existing networking support already available within IP protocol software stacks and operating systems, thereby reducing development and deployment costs, and simplifying redesign and replication efforts. Software modifications and upgrades are also easily performed. Third, many wireless deployment applications require an IP routing approach at some level of the architecture, so improvements relating to wireless performance, mobility, and robustness are generally desirable.

9.1.2 What Do MANET Nodes and Networks Look Like?

MANET-enabled devices can come in a large variety of forms, but there is basic functionality that all MANET nodes at least partially possess. Figure 9.3 shows two types of MANET nodes: first, a simple MANET node is shown as a locally contained computer host with a single wireless network interface; second, a more connected MANET node is shown supporting multiple wireless or wired interfaces and potentially connected network segments. This second node is also potentially providing direct routing support for a set of attached hosts or network prefixes as well.

The overall functional description of a MANET wireless routing node is largely consistent with the conventional view of Internet routing devices (see Figure 9.3B), except for some interface behavioral design assumptions. One such difference is that a *single* interface device is acceptable as a common form of a MANET routing node (See Figure 9.3A). This is true because in a broadcast-oriented wireless interface, a node *A* will have

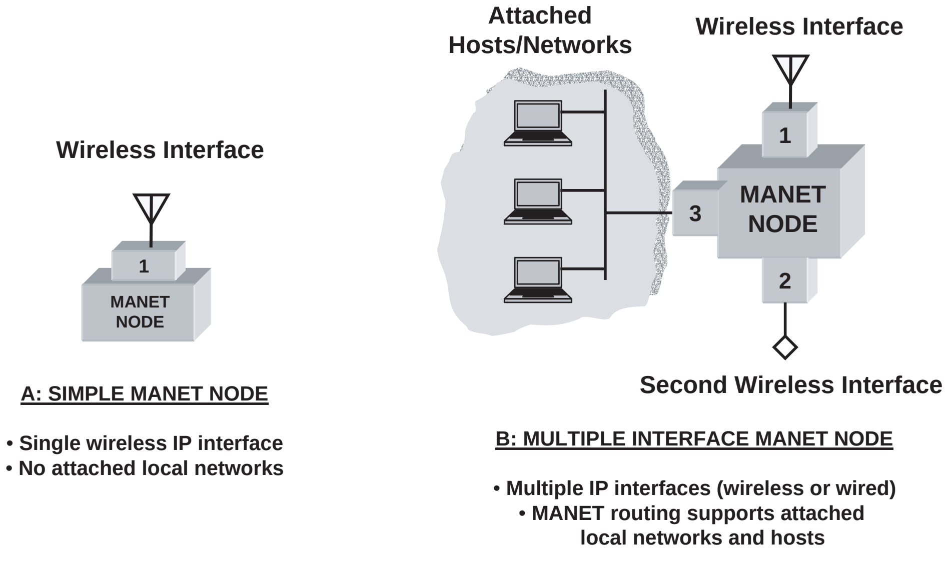

Wireless Interface
1
MANET NODE
A: SIMPLE MANET NODE
• Single wireless IP interface
• No attached local networks
Attached Hosts/Networks
Wireless Interface
1
3
MANET NODE
2
Second Wireless Interface
B: MULTIPLE INTERFACE MANET NODE
• Multiple IP interfaces (wireless or wired)
• MANET routing supports attached local networks and hosts

Figure 9.3. Types of MANET node configurations.

neighbors on the common physical interface that another node *B* may not have, due to relative node positions or other interference and propagation effects. Thus, unlike many wired protocol designs, individual nodes may and often should forward downstream traffic on the *same* interface on which a data or control packet was received. This forwarding is required on behalf of other nodes out of range of a sending upstream node. There are other scenario-dependent factors that make MANET network interfaces potentially different from conventional wired network interfaces as follows:

- Relative neighbor motion, environmental, and distance effects
- Dynamic local noise and interference (possibly self-induced)
- Time-varying communication channels due to intentional or unintentional causes
- Asymmetric neighbor links often exist
- Lower-layer wireless protocol behaviors (retransmit buffers, reliability, etc.)

The main point here is that MANET neighbor interface links can and often do behave and appear quite different than those assumed in the wired networking world.

In addition to enhanced wireless mobile routing, connectivity and interoperability with the rest of the Internet infrastructure is an important area of consideration. At present, working MANET experimental networks often operate as Internet stub routing areas.

We now consider how a set of deployed MANET nodes might appear within a larger network context. Once again, there exist a large variety of possible application and deployment scenarios, and we discuss only a few general examples. In its present stage of engineering experience, MANET technology is perhaps most commonly suitable as a means to provide Internet connection for a wireless routing region at the *edge* of the Internet—to extend network range or to provide robust adaptation to infrastructure dynamics. Figure 9.4 illustrates a MANET stub area connected through a single Internet access router. In this case, the MANET routing is functioning solely at the edge of an existing Internet infrastructure to support the seamless extension of that infrastructure into a more dynamic, wireless environment. Existing, low-complexity MANET routing solutions that have recently emerged from research into development can be used effectively within

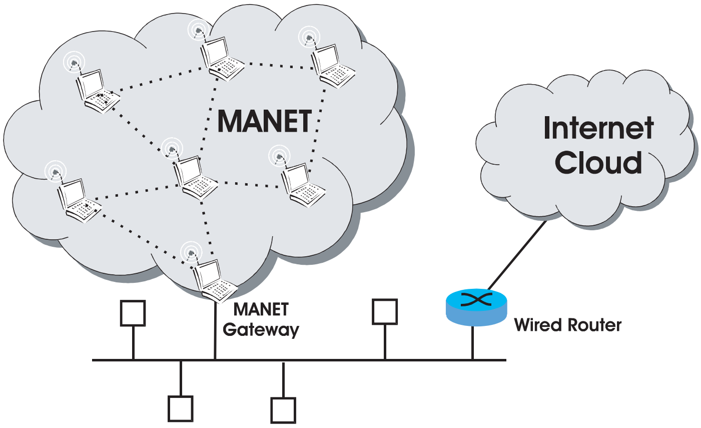


Figure 9.4. Basic MANET Internet extension.

such a stub network scenario. In addition, multiple Internet points of attachment can often be supported but are not shown in Figure 9.4. Since, in this scenario, connection and access to the greater Internet is assumed to be of primary importance, addresses within the area can be managed and even autoconfigured through protocol and management techniques focused at the edge of the MANET area.

Stub area operation simplifies routing interoperability issues and forwarding policies and is often the most typical application scenario of present MANET experimentation and deployment. Recent proliferation in working MANET-based experiments and demonstrations has been made possible by the variety of working prototype implementations available for general use [8]. The use of a globally advertisable prefix or set of prefixes within the MANET stub is quite straightforward and simplifies the router border gateway advertisement and routing table exchange issues. This stub scenario also allows simpler approaches to MANET autoconfiguration, and at the Naval Research Laboratory (NRL) we have demonstrated a number of working small-scale networks demonstrating MANET autoconfiguration, as depicted in Figure 9.5.

In the working example, node A serves as a MANET gateway node and provides default routing to the larger Internet. Node X is a wireless node that has been previously autoconfigured and is now operating as a functional MANET router and simultaneously as a Dynamic Host Configuration Protocol (DHCP) relay agent for node A. A joining node Y now enters the network and obtains a MANET region routable address via associated neighbor DHCP relay agent functionality. Once it obtains a valid routable address, node Y can begin participating in a MANET routing protocol and also begin hosting related services such as a DHCP relay function. At NRL, we are also demonstrating related autoconfiguration operations for IPv6-oriented wireless networks and are looking at the integration of stateless address autoconfiguration and distributed service discovery and interaction methods involving anycasting. Our work and related work of others is presently ongoing but will likely result in multiple approaches and techniques to autoconfigure MANET networks in the future.

As demonstrated in Figure 9.5, a small-scale MANET can be completely autoconfigured including the establishment of addressing, service discovery, and dynamic MANET

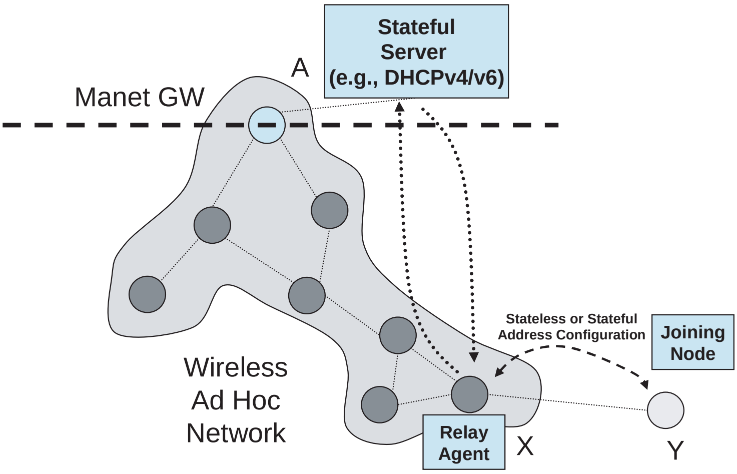


Figure 9.5. MANET stub autoconfiguration example.

routing. By using simple conventional-type approaches for autoconfiguration combined with a MANET routing protocol, an adaptive wireless network can be realized for many practical applications (e.g., a routable home network). More complex MANET address management and configuration schemes can be devised, and may be of interest, but are not discussed further here since these concepts are evolving and they may not be necessary for many practical applications.

Beyond the more common Internet extension application for MANETs, dynamic MANET routing areas may also be formed independent of any external network connection to support completely *autonomous* networked operation. Examples include the use of such techniques within an area to perform ad hoc collaboration and computing in an emergency or disaster relief scenario. Other such uses could be in temporary, dynamic collaborative business, private intranet, or robotic applications. In the case when an infrastructure connection is lacking or not of primary application interest, the autoconfiguration and addressing issues become more challenging. In this case, more peer-to-peer type approaches may be useful and are still being discussed and evolved within the technical community.

Only a few years ago, few actual working implementations of MANET protocols existed in practice. Yet, at the time of this writing, a wide variety of MANET prototype software has been demonstrated on a variety of diverse end platforms. A key point relating to this is that interoperating MANET nodes can always be a heterogeneous collection of devices and platforms of different capabilities and uses. Figure 9.6 shows a number of different sublaptop to handheld devices presently demonstrating functional MANET routing capabilities within a dynamic wireless topology.

9.1.3 Wireless Characteristics and Applicability

The goal of MANET routing is to provide enhanced IP routing for wireless networks, especially those that are possibly mobile or highly dynamic. The lessons learned from such designs may also be useful in wired protocols, but the unique challenge of wireless operation provides the primary design motivation. We previously mentioned a few unique behavioral aspects of wireless MANET interface types. In addition, there are numerous operational factors that significantly distinguish mobile wireless networks from fixed networks including [3]:

- Nominally *lower capacity* is typically available as compared to wired network counterparts. This is becoming less of a concern in more recent applications using short-

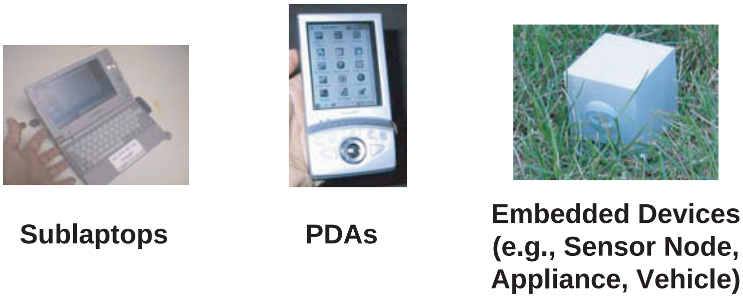


Figure 9.6. Some Example MANET prototype platforms.

range, high-capacity wireless communications, but there remain scenario-dependent issues relating to power, spectrum, and antenna design.

- *Limited broadcast* nature of some wireless multiple access media. Again, this relates to the use of a single interface for relaying and connecting devices out of range but on the same physical interface. Many existing wired routing protocol designs assume that this type of forwarding should not occur.
- Increased likelihood of *channel interference* and *congestion detection* problems. This may be due to bandwidth constraints, hidden-terminal problems, frequency restrictions, or channel access techniques.
- More *frequent topological changes.* This may often be due to node mobility, channel propagation effects, resource failures, power control, or antenna dynamics.
- *Higher loss rates* (e.g., due to interference, fading, congestion or network dynamics)
- Potentially *higher delays and jitter* (e.g., due to lower transmission rates, link layer retransmissions, use of long propagation delay links, or dynamics)
- *Lower physical security* of media (e.g., due to lack of physical control over media)

There is a significant history of packet radio network research and development going back to the early 1970s [14]. However, in the past, mobile wireless network designs were often looked upon as *homogeneous* radio frequency (RF) media problems. With time and the proliferation of numerous proprietary radio networks, the need for *heterogeneous* interoperability across networks is becoming a pressing concern. Reflecting upon past Internet technology development, it is clear that support for a heterogeneous mix of technologies and devices is one of the great successes of IP. In the near future, computing and network routing devices may typically have multiple wireless media interfaces (e.g., ultra-wideband, Bluetooth, Zigbee, 802.11 variants, cellular). This proliferation of ubiquitous wireless devices is expected to continue to evolve with many newer technologies to choose from over time. IP routing-layer technology provides multihop relaying and dynamic internetwork connection support. In this broad sense, IP technology has supported and will continue to support both wired and wireless infrastructures.

As we face increasingly embedded and widespread wireless network technology, wireless routing performance that deals with increasing temporal and topological dynamics is a key enabler. We wish to emphasize that increased dynamics may *not* always result from mobility, and, therefore, mobile systems are not the only context suitable for applying MANET technology. Dynamic link conditions due to other system effects are often quite evident in wireless networks, even when the nodes are static or quasistatic. In fact, MANET approaches will likely work equally well or better than existing standard routing when used in quasistatic wireless applications involving routing meshes. Dynamics, without significant motion, may be expected in deployed cooperatives or community network grids, where nodes may come or go at random, or in networks where energy conservation or power cycling may be an issue. Regardless of the operational reason, the ability to *manage and adapt to expected change* with a high degree of robustness and efficiency is assumed to be a fundamental desired property.

9.1.4 Networking with Small Devices

MANET technology, because it provides dynamic and mobile wireless network support, is naturally being considered for use in embedded devices, including human wearable and portable devices. As illustrated in Figure 9.6, devices such as compact laptop computers, personal digital assistants (PDAs), and embedded computing systems have been demonstrated running varieties of MANET routing technology. This technology achievement enables the ad hoc formation and maintenance of dynamic wireless infrastructures even among small, embedded devices. However, there are a number of issues related to the use of small, portable devices that deserve consideration when developing and selecting appropriate MANET technology or modes of operation including:

- More limited energy (e.g., possibly battery-/solar-operated devices)
- More limited computing power
- More limited memory
- Increased interference and dynamics (nearfield RF effects)

Limited energy can be a critical operational consideration. For instance, if a battery-powered device is naturally power cycling (e.g., into and out of a dormant mode) this may need to be considered as an effect in the design assumptions of a routing protocol. Questions of importance may include the following:

- What are the related energy costs of transmit, receive, and dormant modes of a node?
- Is transmit power control desirable to adjust the number of active neighbors in a topological region?
- Is it desirable to conserve the energy of the network as a whole, the individual device, or both?

Besides the potential energy conservation issue with embedded devices, there is also an issue of complexity. A heavyweight, complex routing protocol that may work well on a modern laptop or desktop computer with significant memory and processing capability is not necessarily the best approach for an embedded network processor or PDA application. The fact that embedded computers vary in their memory and computing capabilities—some rivaling desktop systems of several years ago—lead us to several conclusions on this issue. First, there is likely more processing and memory freedom in algorithm and protocol design than there was in the design space of the early Internet days, even for many embedded computing applications. Yet, second, while Moore's Law seems to equally apply to embedded computing capability growth (albeit as a lagged variant), we still need to be cautious as there is an increasing cost to be paid for burdening these precious local resources with overly burdensome protocol and software designs, especially since battery capacity improvements are not following Moore's Law.

9.2 MANET TECHNOLOGY APPLICATION

This section discusses more select scenarios that the authors are aware of being demonstrated successfully using MANET technology and future applications that are being considered for adaptation and use. This list of uses and applications is not intended to be complete and there are other applications for MANET technology. Likely, there are many that are not presently realized or envisioned by the authors, not unlike the case of the early Internet. To begin, we reemphasize that MANET is *not* solely intended for disconnected autonomous operation or scaled scenarios (e.g., hundreds or even thousands of cooperating wireless nodes in a region). These are interesting and important potential application areas, but we should not ignore the important and more practical basic infrastructure enhancement applications of MANET. As an example, the general use of MANET technology as a basic wireless stub network extension, as illustrated in Figure 9.4, has many potential common applications and for small-to-moderate-size network scenarios is a highly practical approach. Beyond this basic application, we now discuss a set of other potential application areas and related issues.

9.2.1 Hybrid Infrastructure Extension

In the authors' opinion, a potentially widespread use of MANET solutions is likely to be in supporting low-complexity, effective hybrid infrastructure extensions where needed or desired. Many MANET solutions are low complexity and are beginning to become available in a wide variety of implementations. A simple example of an application that the authors have actually deployed on a small scale is a dynamic enhancement to a home or campus wireless networking environment. A typical campus deployment process today involves wireless area surveys to decide how many access points to deploy and where to deploy them. This can work reasonably well if network wireless access devices are deployed, managed, and configured properly to cover all communication areas of interest and all possible scenarios of desired network communications. Some node handoff techniques are designed for distributed access points, but more direct dynamic IP routing and IP router device association and flexibility can offer advantages over proprietary and more limiting techniques within an operational region. Typically, even within a well-architected fixed-backbone system there are problem coverage spots, dynamic outages, as well as short term and long term-dynamics that should be addressed with more flexibility. The use of MANET technology to provide extended service allows low-cost, low-complexity dynamic adjustments to provide coverage regions and range extensions away from the more fixed infrastructure backbone networks.

More recent MANET protocol developments support options to allow certain MANET routing nodes to be preferred over others in a neighborhood [15, 16]. In the extreme, a node can participate in dynamic neighborhood discovery mechanisms of a MANET protocol but may be excluded through low preference from becoming a forwarding or routing node. Through the use and management of a such functions, a fixed wireless infrastructure (preferred and close to the Internet backbone) can be deployed in a dynamic environment with more passive MANET nodes participating in local neighbor exchanges and discovery only. If management policy or preference allows, these more passive MANET nodes can provide range extension and dynamic routing functions on an as-needed basis. A fictitious but somewhat practical example of such a hybrid grid deployment is shown in Figure 9.7, where the static building and utility pole nodes are managed wireless MANET devices that have been well placed and/or loosely coordinated to form a wireless access grid. The wireless access grid in this example is made up of preferred-access nodes close to the fixed infrastructure. Other nodes in the picture demonstrate passive MANET nodes that may be static, power cycling, or moving through the grid to gain Internet communications. In this way, nodes moving or operating on limited energy may be low-preference routing nodes, thus providing more physical stability to the overall routing grid as well. Also depicted is a node at the bottom right that cannot reach or discover any primary access grid nodes and requires some range extension assistance. In this case, a previous passive MANET node, to the left of the one shown in the lower right of Figure 9.7, is provid-

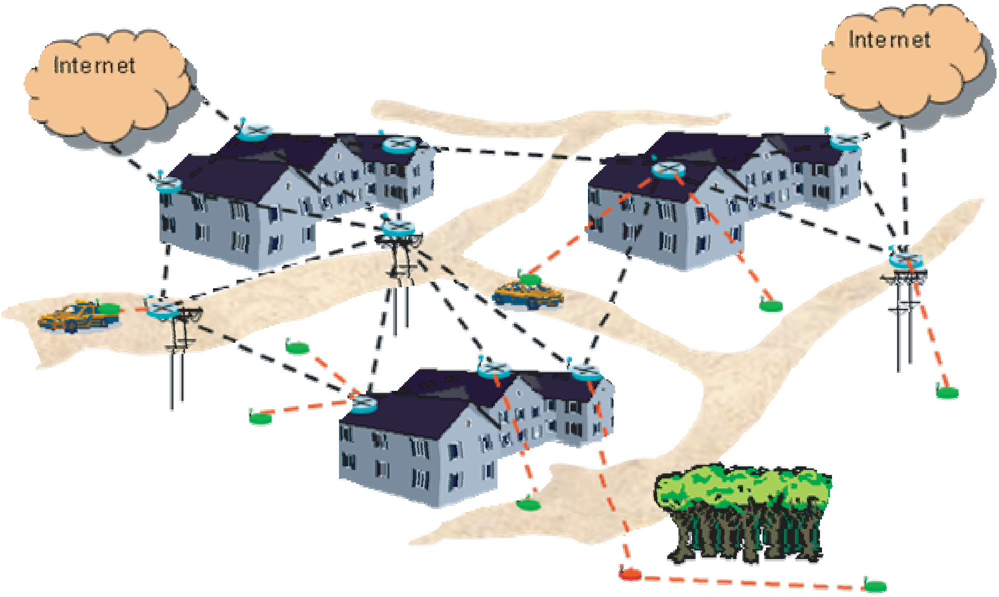


Figure 9.7. Hybrid MANET access grid application.

ing a limited routing function (if policy allows) for the lower right node that is out of range of any preferred nodes. When an assisted passive node can directly link to a primary access router, the assisting node can quickly return to its passive role within the grid. The additional routing functionality to support such a flexible capability is very lightweight but can provide a powerful management function for hybrid systems.

The hybrid grid notion demonstrated by Figure 9.7 illustrates a very practical application of MANET technology that may be appropriate within a campus, community, robotic, sensor, or localized business application. The passive MANET concept easily supports network devices that require dynamic routing support but are not preferred or allowed routers (e.g., intermittent battery-powered PDA connections, known highly mobile or disadvantaged nodes, etc).

It is in contexts such as hybrid applications that MANET technology can most effectively contribute toward the oft-cited vision of "Ubiquitous Computing," a research field originated by Mark Weiser and described in his seminal paper [1]. To quote from the paper, "The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it." One of those technologies is computing. Computing will become truly ubiquitous when connectivity is seamless and its transparency assumed. Self-organizing, adaptive network technologies like MANET are a necessary component—a step toward the realization of Weiser's vision.

9.2.2 The Case of Fixed Infrastructureless Operation

Completely fixed infrastructureless operation is another important potential aspect of MANET technology and can support a wide variety of present and future envisioned applications. In a completely untethered communication scenario, there may be no connection to the greater Internet, but participating users may wish to form a cooperative infrastructure. Within this infrastructure, cooperative nodes can share data and services, and potentially perform a host of other tasks. Ad hoc conferencing or business meeting capabilities and ad hoc homeland defense and disaster relief networks are examples or the more esoteric ad hoc applications. A MANET operating in an autonomous fashion from the Internet or a greater network infrastructure likely has no fixed access nodes or gateway points to provide configuration coordination, so such networks will likely require more distributed forms of autoconfiguration, service discovery, and management. The rapidly evolving area of peer-to-peer technologies may be highly appropriate for these sorts of scenario applications as well, given the likelihood that there will be no presumed infrastructure hierarchy. Participating devices will likely have to operate in a peer-to-peer fashion with appropriate applications and protocols, or nodes will have to dynamically elect and discover distributed services as the network region becomes operational and as nodes with different capabilities and missions join and leave. Distributed election strategies may also play a role in more robust operations of distributed services where required in these scenarios.

9.2.3 Other Network Application Areas (Cooperatives and Sensors)

A new area of growing Internet access and deployment interest around the world is in the area of cooperative, community-based networking. In this case, a community of interest (e.g., small town government, infrastructure-lacking world region, group of interested individuals/club) could own and operate a network infrastructure in a cooperative fashion, much as local roads supplement larger highway infrastructures. If such networks deployed MANET technology to support a self-organizing, adaptive infrastructure, it would likely be similar to the hybrid infrastructure extension application we described previously. One promising example is that of disadvantaged rural regions or developing countries that may be able to build and operate inexpensive, network infrastructure services with the help of MANET technology. Such regions often lack the resources or environment suitable for attracting significant fixed-infrastructure developments and services.

More capable and scaled sensor network applications are of growing interest both for commercial, environmental, and military applications. MANET technology is being considered as one technology component to support broad applications of self-organizing and distributed sensor networks. As with generic communication applications, the advent of small, inexpensive, low-power sensing and computing devices results in novel opportunities for networked sensor applications. Large-scale deployment of networked sensors for a variety of applications is becoming more feasible, both technically and economically. As in our previous examples, both more systematic (e.g., hybrid grid) and ad hoc deployments of sensor networks are of interest. Not all sensor network applications will desire to use Internet and MANET technology, but many applications will want to take advantages of adaptive self-organization and the benefits of building upon the IP design suite. This is especially true as the role of sensor devices becomes more programmable, adaptive, and cooperative in a network setting. Developments in this application area are likely to continue to grow in interest and maturity over the coming years.

9.2.4 Large-Scale versus Small-Scale Use

MANET technology has been envisioned for deployment in many different application scenarios. We have discussed a few broad related application areas in this chapter. A clear distinction needs to be made between simple, self-organizing networks operating on small scales (standalone, moderate-size stub networks, or limited hybrid extension grids), and those expected to perform adequately in regions with a large set of peer routing nodes with every node acting as an active router. Large-scale ad hoc networks face a myriad of difficulties. Conceptually, the core problems faced—sensing and adjusting to dynamic conditions—are the same on small and large scales. But in practice, networked communication dynamics favor smaller wireless networks or at least networks requiring fewer hops in order to access a fixed backbone. For example, in ad hoc networks employing omnidirectional antennas, recent research [6] has shown that as the network scales (assuming it is equally likely that any pair of network nodes wish to communicate) as the number of communicating network nodes grows, the achievable throughput per node pair goes to zero. This analysis does not even consider the bandwidth required to run a routing protocol. As such, the prospect of deploying very large-scale ad hoc networks based on broadcast transmissions is not very promising if the level of nonlocal communication is nonnegligible.

More promising, perhaps, in large-scale deployment of MANETs is operating in a *hybrid* fixed/ad hoc grid fashion, on the edge of a fixed, high-capacity network. In such deployments, traffic destined to topologically proximate MANET nodes would be forwarded directly in the MANET cloud, whereas traffic destined for topologically distant MANET nodes would be forwarded to a MANET/fixed network gateway near the sender, then through the fixed network to a similar gateway near the destination, and, finally, through the MANET cloud to the destination. There are many ways to envision realizing such hybrid fixed/MANET interoperability and we discussed some of the routing aspects of a scenario depicted in Figure 9.7.

One possibility is for IPv4 MANET nodes to use as their host-routed interface address in the MANET network an address that is also a Mobile IP (MIP) Home Address (HoA) registered in a fixed network MIP Home Agent (HA) with reachability to all relevant MANET gateways. If a given IP address does not appear in a MANET node's routing table, its default forwarding behavior is toward the nearest MANET gateway (likely via some form of MANET anycast routing). Such packets arriving at the gateway are forwarded to the HA, and then tunneled to the gateway near the MANET destination node. This MANET gateway is also a Mobile IP Foreign Agent (FA) that advertises its Care-of Address (CoA) locally within the MANET via topologically scoped flooding using a MANET broadcast service. The MANET destination [also a Mobile IP Mobile Node (MN)], having learned this CoA information, would have registered its HoA-CoA binding with its HA via the local MANET Gateway/MIP Foreign Agent. Upon reception of the tunneled packet, the FA will decapsulate the packet and natively route (via MANET routing) the packet to the MANET destination node. In this fashion, MANETs of literally any size may be deployed on the edge of a fixed network infrastructure.

Many other interoperability techniques can be and have been proposed as well, including approaches for IPv6. The key consideration is that packets be routed off the MANET cloud *as soon as possible.* This is because each MANET hop is expensive from a communication perspective relative to fixed network bandwidth. In fact, due to the numerous complications present in mobile wireless networks, it is generally desirable to use the fixed network bandwidth whenever possible. MANET technology, generally presenting the most stressing wireless networking environment, should typically be applied where and when the convenience and dynamic networking opportunities afforded by MANET technology is worth the potential likelihood of reduced performance.

9.3 MANET AND RELATED IETF WORK

The Internet Engineering Task Force (IETF) is a large, open international community of network designers, operators, vendors, and researchers concerned with the evolution of the Internet architecture and the smooth operation of the Internet. The principal products of the IETF are Internet Standards, generally published in the format of Request for Comments (RFC) documents. The actual technical engineering work of the IETF is largely done in its working groups, which are organized by topic into several areas.

Although there are many past and present technical efforts dealing with some aspect of network mobility, an IETF effort that has considered wireless mobility from its inception in the late 1990s has been the MANET Working Group (WG). The MANET WG's primary purpose is to focus on dynamic, wireless IP routing technology and develop and evolve MANET-related routing specifications and introduce them to the Internet Standards process. From its inception, the WG has been acutely aware that network operation in dynamic wireless environments challenges present requirements and design assumptions of Internet protocols and applications—requirements and assumptions derived in the past from a largely *wired and tethered* network mindset. That is *not* to say that applications should be designed specifically for use in wireless networks. Rather, applications and protocols should be designed to gracefully accommodate changes and degradations in connectivity, changes that may occur more frequently in wireless networking contexts, but which occur in fixed networks as well due to congestive dynamics and other factors. Prior to its current status as a topic of Internet standards work, MANET technology had a long history.

9.3.1 History and Motivation

The technology of MANET is somewhat synonymous with Mobile Packet Radio Networking (a term coined during early military research in the 1970s and 1980s), Mobile Mesh Networking [2] (a term that appeared in an article in *The Economist* regarding the structure of future military networks) and Mobile, Multihop, Wireless Networking (perhaps the most accurate term, although a bit cumbersome). Since its inception at the 39th IETF in Munich in August 1997, the MANET WG functioned in a part research, part engineering mode, and has developed and fostered a significant number of proposed routing protocol and analysis methods. The WG has continued to evolve and recently is planning to convert its scope from the pseudoresearch mode of its origin into a pure engineering group, with more mature technical work proceeding through group consensus and scoped near-term problem areas. In parallel with this rechartering, an Internet Research Task Force (IRTF) MANET Research Group was created as a subgroup of the IRTF Routing Research Group. The research theme of this research group is to concentrate on more complex scalability issues and concerns. Several IETF MANET Internet Drafts, and two basic reactive and two basic proactive protocol designs have reached a reasonable level of maturity, analysis, and implementation experience. These include the following:

- Ad Hoc On-Demand Distance Vector (AODV) [18]
- Distributed Source Routing (DSR) [17]
- Optimized Link State Routing (OLSR) [16]
- Topology Dissemination-Based Reverse Path Forwarding (TBRPF) [15]

Work on the design of the protocols AODV, DSR, OLSR, and TBRPF is roughly completed and these protocols will likely be soon considered for Experimental RFC status to encourage additional third-party (e.g., industry) experimentation with concepts and implementations. Lessons learned from the past development of these core protocols will form an engineering basis for any follow-on efforts of more consensus-based protocol work in the future. The MANET IETF WG has produced a significant body of prototype MANET protocols, but work on developing standards will continue to evolve as more is learned from actual application experience and appropriate Internet engineering efforts in the coming years.

9.3.2 Some Future Work Issues

Those familiar with wireless network design often realize that layered design is more complicated than in wired networks due to more severe functional and performance dependencies between the layers. Thus far, the Internet has been designed, and its protocols optimized, largely for fixed-network deployment scenarios. Mobility is still largely a second-class citizen in the Internet standards, and its support has been mostly an afterthought. Even "Mobile" IP was originally intended to support portability; i.e., the ability to connect on a foreign subnet using a local address (a colocated care-of address) as a source address, and yet remain reachable via a second "home" address, possibly stored in the dynamic name system. Larger system aspects as they relate to mobility, such as handoff, were seemingly not originally considered.

The dominant wired network design tradition translates directly to the types of interfaces that IP expects to support, and the dynamics assumed to be associated with those interfaces. Layer 2 interfaces fundamentally operate as either broadcast or point-to-point. From these primitives, additional Layer 3 interface constructs such as nonbroadcast multiple access and point-to-multipoint are created as necessary. This approach has served the wired Internet well. However a third type of Layer 2 interface is necessary to efficiently and seamlessly extend IP over dynamic networks, principally wireless ones. This interface, here termed a "dynamic" interface, combines traditional broadcast interface addressing semantics (i.e., support for unicast, multicast, and broadcast link-layer addresses) with Layer 2 association event support for the dynamic creation of peer-to-peer interface associations within an otherwise broadcast interface. Its intended domain of applicability covers cellular, WLAN, MANET, and so on; in short. all currently envisioned forms of dynamic wireless networking. The support for this form of interface in all IP stacks would enable Layer 2 designers to craft link layers that transfer standardized signals to an IP stack. Simple event notifications such as link active/inactive would facilitate efficient IP/link layer interaction, without the need for expensive, periodic IP-level signaling (e.g., Hello Beacons) to ascertain IP-layer neighbor information. Moreover, with such functionality standardized in IP's underbelly, future link layers may be developed with IP-awareness such that the link active event could immediately convey a neighboring IP address to an adjacent stack. The IP stack can choose to do what it wishes with such information, including ignoring it if it is so configured.

As the understanding of network and wireless dynamics increases, quantitative information such as link quality can also be normalized into metrics suitable for consumption by a dynamic interface-aware IP stack, and conveyed to a routing protocol. It is an open question as to how much integration of information sharing should occur between a network and lower layers. But beginning even basic work in this direction supports a move toward more optimal design strategies, and good solutions can only improve MANET performance over the present state of the technology.

Context-aware routing is a promising area for future MANET research. The idea is that the network routing protocol itself can be *polymorphic,* and be able to adapt its operation to a changing network membership and deployment context in real time. Realize that the network itself may be moving and its composition changing. The most appropriate routing policy may change during the lifetime of a network or, even more interestingly, certain contiguous regions of a MANET may choose to run different routing policies at the same time, with MANET routers along the border of different regions serving as dynamic routing protocol gateways.

Providing Quality of Service other than best-effort or simple Differentiated Services is a very difficult problem in MANETs. Recalling the previous discussion on integrated design, much of what is achievable in terms of QoS depends in large part on the link-layer technology in use. The extent that the particulars of any given link layer can be effectively conveyed to an IP standard routing protocol in a standardized yet useful fashion remains to be seen. This is a challenging area of future research. Finally, perhaps the biggest challenge facing MANET QoS is that many of the link layers responsible for its popularity run in the unlicensed spectrum. It is simply infeasible to provide strong QoS guarantees in a spectrum you do not control.

9.4 CONCLUSION

Although significant packet radio work was begun in the 1970s, recent growing interest and associated technology revolution of MANET wireless technology has fundamentally resulted from two recent technological events:

- The *everywhere in everything* Internet Protocol (IP) revolution
- The *embedded, inexpensive, unlicensed* wireless technology revolution

Dynamic, wireless IP-based routing has reached a level of practicality and maturity for a wide variety of scenarios. We discussed four practical, near-term applications for MANET technology that we feel provide needed capability enhancements for today's evolving wireless network applications:

1. Small-to-moderate-sized Internet stub wireless routing regions (see Figure 9.4)
2. Small-to-moderate-sized autoconfiguring wireless networks (see Figure 9.5)
3. Hybrid MANET-enabled access grid regions, handling dynamics yet providing efficient fixed Internet backbone access (see Figure 9.7)
4. Untethered, ad hoc collaborative networks

We briefly discussed related MANET work that has recently been accomplished within the Internet Standards forum and the potential future direction for MANET work in several key areas. The understanding gained from these early activities will now be used to develop standardized MANET routing functionality for the Internet. More general lessons learned in terms of improved robustness and efficiency may also help influence and improve future designs of wired protocols. Despite recent development progress and confidence in some deployment scenarios, there still remains much to do in terms of understanding, developing, and deploying a broad base of MANET technology. One promising area of future work is engineering improvements in extending network and wireless interface design to improve system effectiveness. This is also a problem area for other wireless, mobile Internet efforts and is not limited to routing issues.

Overall, significant progress in the area of MANET technology has been realized in recent years. After years of research and development, we are now seeing a growing sense of widespread, practical applications for this technology along with the more esoteric visions. We anticipate further proliferative growth of wireless network systems at the edge of the Internet infrastructure and, therefore, more requirements for adaptive, flexible, and robust operational concepts in the next decade. As our daily lives and our societies begin teeming with wireless devices, *everywhere and in everything,* adaptive network technologies will further enable the envisioned, invisible computing world.

REFERENCES

1. M. Weiser, "The Computer for the Twenty-First Century," *Scientific American,* pp. 94–10, September 1991.
2. M. S. Corson, S. Batsell, and J. Macker, "Architectural Considerations for Mobile Mesh Networking," in *Proceedings IEEE MILCOM* 96.
3. J. P. Macker and M. S. Corson, "Mobile Ad Hoc Networking: Routing Protocol Performance Issues and Evaluation Considerations," Internet RFC 2501, Jan 1999.
4. M. S. Corson, J. P. Macker, G. Cirincione, "Internet-Based Mobile Ad Hoc Networking," *IEEE Internet Computing, 3,* 4, July/August 1999.
5. J. P. Macker, V. Park, and M. S. Corson, "Mobile and Wireless Internet Service: Putting the Pieces Together," *IEEE Communications Magazine,* June 2001.
6. P. Gupta and P. R. Kumar, "The Capacity of Wireless Networks," *IEEE Transactions on Information Theory, IT-46,* 2, pp. 388–404, March 2000.
7. C. Perkins et al., *Ad Hoc Networking,* Addison-Wesley, 2001.
8. Partial list of protocol implementations, http://www.wikipedia.org/wiki/Ad_hoc_protocols_implementations.
9. The Network Simulator ns-2, http://www.isi.edu/nsnam/ns/.
10. Qualnet Simulator, http://www.scalable-networks.com/.
11. OPNET Simulator, http://www.opnet.com/.
12. ETSI STC-RES10 Committee. Radio equipment and systems: HIPERLAN type 1, functional specifications ETS 300-652, ETSI, June 1996
13. A. Ephremides, J. E. Wieselthier, and D. J. Baker, "A Design Concept for Reliable Mobile Radio Networks with Frequency Hopping Signaling," *Proceedings of IEEE. 75,* 56–73, January 1987.
14. R. Kahn et al., "Advances in Packet Radio Technology," *Proceedings of the IEEE 66,* 1468–1496, November 1978.
15. B. Bellur and R. G. Ogier, "A Reliable, Efficient Topology Broadcast Protocol for Dynamic Networks," *Proceedings IEEE INFOCOM '99,* New York, March 1999.
16. T. Clausen, P. Jacquet, A. Laouiti, P. Muhlethaler, A. Qayyum, and L. Viennot, "Optimized Link State Routing Protocol," *IEEE INMIC 2001.*
17. D. B. Johnson, D. A. Maltz, and J. Broch, "DSR: The Dynamic Source Routing Protocol for Multihop Wireless Ad Hoc Networks," in *Ad Hoc Networking,* C. E. Perkins (Ed.), pp. 139–172, Addison-Wesley, 2001.

18. C. E. Perkins and E. M. Royer, “The Ad Hoc On-Demand Distance-Vector Protocol (AODV),” in *Ad Hoc Networking,* C. E. Perkins (Ed.), pp. 173–219, Addison-Wesley, 2001.

19. Z. Haas and M. Pearlman, “Zone Routing Protocol (ZRP): A Framework for Routing in Hybrid Ad Hoc Networks,” in *Ad Hoc Networking,* C. E. Perkins (Ed.), pp. 221–253, Addison-Wesley, 2001.

CHAPTER 10

ROUTING APPROACHES IN MOBILE AD HOC NETWORKS

ELIZABETH M. BELDING-ROYER

10.1 INTRODUCTION

Routing in ad hoc networks has become a popular research topic. Dating back to the early 1980s, there have been a large number of routing protocols designed for multihop ad hoc networks. These protocols cover a wide range of design choices and approaches, from simple modifications of Internet protocols, to more complex multilevel hierarchical schemes.

Many of these routing protocols have been designed based on similar sets of assumptions. For instance, most routing protocols assume that all nodes have homogeneous resources and capabilities. This includes the transmission ranges of the nodes. Also, bidirectional links are often assumed. In some instances, protocols have mechanisms for determining whether links are bidirectional. In these cases, the protocols will then eliminate unidirectional links from consideration for routing. In other instances, protocols can actually utilize these unidirectional links, whereas other protocols simply assume all links are bidirectional. Finally, although the ultimate end goal of a protocol may be operation in large networks, most protocols are typically designed for moderately sized networks of 10 to 100 nodes.

Before describing the types of approaches and example protocols, it is important to explain the developmental goals for an ad hoc routing protocol so that the design choices of the protocols can be better understood. As has already been stated in previous chapters, the defining characteristics of ad hoc networks include resource-poor devices, limited bandwidth, high error rates, and a continually changing topology. Among the available resources, battery power is typically the most constraining. Hence, the following are typical design goals for ad hoc network routing protocols:

Mobile Ad Hoc Networking. Edited by Basagni, Conti, Giordano, and Stojmenovic.
ISBN 0-471-37313-3 © 2004 Institute of Electrical and Electronics Engineers, Inc.

- **Minimal control overhead.** Control messaging consumes bandwidth, processing resources, and battery power to both transmit and receive a message. Because bandwidth is at a premium, routing protocols should not send more than the minimum number of control messages they need for operation, and should be designed so that this number is relatively small. While transmitting is roughly twice as power consuming as receiving [26], both operations are still power consumers for the mobile devices. Hence, reducing control messaging also helps to conserve battery power.
- **Minimal processing overhead.** Algorithms that are computationally complex require significant processing cycles in devices. Because the processing cycles cause the mobile device to use resources, more battery power is consumed. Protocols that are lightweight and require a minimum of processing from the mobile device reserve battery power for more user-oriented tasks and extend the overall battery lifetime.
- **Multihop routing capability.** Because the wireless transmission range of mobile nodes is often limited, sources and destinations may typically not be within direct transmission range of each other. Hence, the routing protocol must be able to discover multihop routes between sources and destinations so that communication between those nodes is possible.
- **Dynamic topology maintenance.** Once a route is established, it is likely that some link in the route will break due to node movement. In order for a source to communicate with a destination, a viable routing path must be maintained, even while the intermediate nodes, or even the source or destination nodes, are moving. Further, because link breaks on ad hoc networks are common, link breaks must be handled quickly with a minimum of associated overhead.
- **Loop prevention.** Routing loops occur when some node along a path selects a next hop to the destination is also a node that occurred earlier in the path. When a routing loop exists, data and control packets may traverse the path multiple times until either the path is fixed and the loop is eliminated, or until the time to live (TTL) of the packet reaches zero. Because bandwidth is scarce and packet processing and forwarding is expensive, routing loops are extremely wasteful of resources and are detrimental to the network. Even a transitory routing loop will have a negative impact on the network. Hence, loops should be avoided at all times.

With these goals in mind, numerous routing protocols have been developed for ad hoc networks. There are far too many proposed routing protocols than can be discussed in this chapter. This chapter describes the characteristics of classes of routing approaches, and then describes the operation of particular routing protocols within those classes. The routing protocols that are described are selected for a number of reasons. They may be popular choices for routing-related research among the ad hoc community; they may, at the time of this writing, be under consideration by the Mobile Ad Hoc Networks (MANET) Working Group [34] of the IETF for standardization; or, they may simply be good, illustrative examples of their particular class of protocol. This chapter does not make comparisons regarding the discussed protocols. There have been many published studies comparing the performance of ad hoc routing protocols in a variety of scenarios. The interested reader should see [8, 15, 16, 23, 32, 53] for comparisons of some of the discussed protocols. Further, the citations for the protocols themselves often contain an evaluation of the protocol, and sometimes a comparison with other ad hoc routing protocols. Finally, a high-level description of each protocol is presented. Consequently, many operational details have been omitted. Additional details of each protocol can be found in its respective citation.

10.2 PROACTIVE APPROACHES

The proactive routing approaches designed for ad hoc networks are derived from the traditional distance vector [35] and link state [38] protocols developed for use in the wireline Internet. The primary characteristic of proactive approaches is that each node in the network maintains a route to every other node in the network at all times. Route creation and maintenance is accomplished through some combination of periodic and event-triggered routing updates. Periodic updates consist of routing information exchanges between nodes at set time intervals. The updates occur at specific intervals, regardless of the mobility and traffic characteristics of the network. Event-triggered updates, on the other hand, are transmitted whenever some event, such as a link addition or removal, occurs. The mobility rate directly impacts the frequency of event-triggered updates because link changes are more likely to occur as mobility increases.

Proactive approaches have the advantage that routes are available the moment they are needed. Because each node consistently maintains an up-to-date route to every other node in the network, a source can simply check its routing table when it has data packets to send to some destination and begin packet transmission. However, the primary disadvantage of these protocols is that the control overhead can be significant in large networks or in networks with rapidly moving nodes. Further, the amount of routing state maintained at each node scales as $O(n)$, where n is the number of nodes in the network. Proactive protocols tend to perform well in networks where there is a significant number of data sessions within the network. In these networks, the overhead of maintaining each of the paths is justified because many of these paths are utilized.

10.2.1 Destination-Sequenced Distance Vector Routing

The Destination-Sequenced Distance Vector (DSDV) routing protocol [46] is a distance vector protocol that implements a number of customizations to make its operation more suitable for ad hoc mobile networks. DSDV utilizes per-node sequence numbers to avoid the *counting to infinity* problem common in many distance vector protocols. A node increments its sequence number whenever there is a change in its local neighborhood (i.e., a link addition or removal). When given a choice between two routes to a destination, a node always selects the route with the greatest destination sequence number. This ensures utilization of the most recent information.

Because DSDV is a proactive protocol, each node maintains a route to every other node in the network. The routing table contains the following information for each entry: destination IP address, destination sequence number, next-hop IP address, hop count, and install time. DSDV utilizes both periodic and event-triggered routing table updates. Every time interval, each node broadcasts to its neighbors its current sequence number, along with any routing table updates. The routing table updates are of the form:

< *destination IP address, destination sequence number, hopcount* >

After receiving an update message, the neighboring nodes utilize this information to compute their routing table entries using an iterative distance vector approach [35]. In addition to periodic updates, DSDV also utilizes event-triggered updates to announce important link changes, such as link removals. Such event-triggered updates ensure timely discovery of routing path changes.

As stated previously, if a node learns two distinct paths to a destination, the node selects the path with the greatest associated destination sequence number. This ensures the utilization of the most recent routing information for that destination. When given the choice between two paths with equal destination sequence numbers, the node selects the path with the shortest hop count. On the other hand, if all metrics are equivalent, then the choice between routes is arbitrary.

DSDV implements two primary optimizations to improve performance in mobile networks. The first is that it defines two types of updates: full and incremental. Full updates are transmissions of a node's entire routing table. Because the size of these updates scales with the size of the network, these updates are performed relatively infrequently. To reduce processing overhead and bandwidth consumption, incremental updates are transmitted more frequently. Incremental updates include only those routing table entries that have changed size since the last full update. Once the number of routing changes becomes too great to fit into a single NPDU, a full update is transmitted during the next update period.

Finally, DSDV also implements a mechanism to damp routing fluctuations. Due to the unsynchronized nature of the periodic updates, routing updates for a given destination can propagate along different paths at different rates. To prevent a node from announcing a routing path change for a given destination while another, better, update for that destination is still en route, DSDV requires nodes to wait a *settling time* before announcing a new route with a higher metric for a destination. The settling time is the average time to get all the updated advertisements for a route. In this way, the node can be sure to receive all routing path changes for a destination before propagating any of those changes. This reduces bandwidth utilization and power consumption by neighboring nodes.

10.2.2 Optimized Link State Routing

The Optimized Link State Routing (OLSR) protocol [14] is a variation of traditional link state routing, modified for improved operation in ad hoc networks. The key feature of OLSR is its use of *multipoint relays* (MPRs) to reduce the overhead of network floods and the size of link state updates. Each node computes its MPRs from its set of neighbors. The MPR set is selected such that when a node broadcasts a message, the retransmission of that message by the MPR set will ensure that the message is received by each of its two-hop neighbors. Hence, whenever a node broadcasts a message, only those neighbors in its MPR set rebroadcast the message. Other neighbors that are not in the MPR set process the message but do not rebroadcast it. Further, when exchanging link state routing information, a node only lists its connections to those neighbors that have selected it as an MPR. That set of neighbors is termed the *MPR Selectors*.

The MPR set for a given node is the set of neighbors that covers the two-hop neighborhood of the node, as shown in Figure 10.1. Nodes learn their set of two-hop neighbors through the periodic exchange of Hello messages. Each node periodically transmits a Hello message that contains a list of all neighbors. Associated with each neighbor is an attribute indicating the directionality of the link to that neighbor. The node is labeled *sym-*

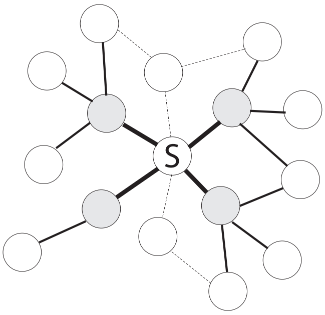


Figure 10.1. Multipoint relays.

metric if the link to the neighbor is bidirectional, or *asymmetric* if a Hello has been received from that node but the link has not been confirmed as bidirectional. When a node receives this Hello message from each of its neighbors, it obtains complete knowledge of its two-hop neighbor set at that point in time. Further, if its own address is listed in the Hello message, it knows the link with that neighbor is bidirectional. It can then update the status of that neighbor to be symmetric.

The MPR set may be calculated according to the following algorithm [27]. Each node starts with an empty MPR set. The set N is defined to be the set of one-hop neighbors with which there exists bidirectional connectivity, and the set N_2 is the set of two-hop bidirectional neighbors. The first nodes that are selected for the MPR set are those nodes in N that are the only neighbors of some node in N_2. Next, the degree of each node n in N that is not in the MPR set is calculated, where the degree is the number of nodes in N_2 that n covers that are still uncovered by the MPR set. As long as there are still nodes in N_2 that are not covered by nodes in the MPR set, the node in N that has the highest degree is included in the MPR set. Once all the nodes in N_2 are covered, this process terminates.

Once each node's MPR set is selected, routing paths within the network can be determined. Because OLSR is a proactive protocol, each node maintains a route to every other node in the network. To diffuse topology information, nodes periodically exchange *Topology Control* (TC) messages with their neighbors. The TC message for a given node lists the set of neighbors that have selected the sending node as an MPR. This is called the *Multipoint Relay Selector* set of the node. Only this set of nodes is advertised within the network. As a node receives TC messages from the other network nodes, it can create or modify routing entries to each node in the network using any shortest path routing algorithm, such as a variation of Dijkstra's algorithm.

10.2.3 Topology Dissemination Based on Reverse-Path Forwarding

A different proactive approach is taken by the Topology-Based Reverse Path Forwarding (TBRPF) protocol [6]. Like OLSR, TBRPF is a link-state routing protocol; however, TBRPF employs a different technique for reducing overhead. TBRPF nodes compute a shortest-path tree to all network nodes. To minimize bandwidth utilization, the nodes then propagate only part of this tree to their neighbors. TBRPF consists of two main modules: a neighbor discovery module for maintaining neighborhood information, and a routing module for topology discovery and route computation.

The neighbor discovery module enables nodes to detect neighors and determine the type of connectivity to each neighbor. Connectivity can be either bidirectional, unidirectional, or, in the case of a broken link, the link can be lost. Each node periodically broadcasts a Hello message to its neighbors. The Hello message is *differential,* in the sense that only changes in the status of neighbors are reported. Each Hello message contains three categories of neighbor information. A neighbor can be listed in the *neighbor request, neighbor reply,* or *neighbor lost* category. The categories aid the nodes in determining the directionality of the links with their neighbors. In general, when a node *i* changes the status of its neighbor *j*, it includes node *j* in the appropriate list (indicating the neighbor's new status) in (typically) three consecutive Hello messages. This ensures that node *j* will either learn of the status change or will declare node *i* to be lost after missing this number of Hello messages.

The neighbor request list contains the addresses of those neighbors from which the node has recently received Hellos, but for which it has not yet been determined that the link to those neighbors is bidirectional. When a node *i* receives a Hello from a new neighbor *j*, node *i* lists *j* in its neighbor table and flags the entry as a unidirectional link. The next time node *i* transmits a Hello message, node *i* lists node *j* in the neighbor request list of that message. This indicates that *i* is requesting that *j* confirm the receipt of *i*'s Hello, so that the bidirectionality of the link can be confirmed. When node *j* receives node *i*'s Hello listing *j* in the neighbor request, node *j* creates a neighbor table entry for *i* (if one does not already exist), and marks the entry as bidirectional. In its next Hello, node *j* includes *i*'s address in the neighbor reply list, indicating that the Hello message from *i* was received and a bidirectional link exists. Node *i* can then update the entry for node *j* to be bidirectional. To prevent the inclusion of transient links, nodes can wait to receive some threshold number of Hello messages from a neighbor before creating a neighbor table entry for that node.

Once a node has created a neighbor-table entry for a neighbor, the node must monitor the status of that link to ensure that connectivity to the neighbor continues to exist. If a node *i* misses the threshold number of Hello messages from a neighbor *j*, it updates the state of that neighbor to be lost. The next time *i* sends a Hello message, it includes node *j*'s address in the neighbor-lost list. If *j* receives the Hello, it notes that the bidirectional connection to *i* has been lost, and it changes the status of node *i* in its neighbor table to unidirectional. Otherwise, if node *j* fails to receive further Hello messages, it deletes node *i* from its neighbor table and includes it in the neighbor-lost list of future Hello messages.

To perform routing, each TBRPF node computes a shortest-path source tree to each reachable node in the network. The tree is computed using a modified version of Dijkstra's algorithm. After computing the tree, nodes report only a part of the tree, called the *reportable subtree* (RT), to neighboring nodes. To report RT, two types of topology updates message are used. Periodic updates are sent reporting the entire RT. These full updates are utilized to inform new neighbors of RT and ensure that all the necessary topology information is eventually propagated. Smaller, more frequent updates are sent as differential updates. The differential updates report only those changes to RT that have occured since the last periodic update. To reduce the number of control messages, topology updates can be combined with Hello messages so that fewer control packets are transmitted.

To calculate RT, let *T*(*j*) denote the subtree of node *i*'s source tree rooted at neighbor *j*. For each neighbor *j*, node *i* includes *T*(*j*) in its reportable subtree RT if and only if it determines that one of its neighbors may select *i* to be its next hop on its shortest path to *j*. To make this determination, node *i* computes the min-hop paths from each neighbor to every other neighbor, using the node identification (ID) to break any ties.

10.3 REACTIVE APPROACHES

Reactive routing techniques, also called *on-demand* routing, take a very different approach to routing than proactive protocols. A large percentage of the overhead from proactive protocols stems from the need for every node to maintain a route to every other node at all times. In a wired network, where connectivity patterns change relatively infrequently and resources are abundant, maintaining full connectivity graphs is a worthwhile expense. The benefit is that when a route is needed, it is immediately available. In an ad hoc network, however, link connectivity can change frequently and control overhead is costly. Because of these reasons, reactive routing approaches take a departure from traditional Internet routing approaches by not continuously maintaining a route between all pairs of network nodes. Instead, routes are only discovered when they are actually needed. When a source node needs to send data packets to some destination, it checks its route table to determine whether it has a route. If no route exists, it performs a *route discovery* procedure to find a path to the destination. Hence, route discovery becomes on-demand. If two nodes never need to talk to each other, then they do not need to utilize their resources maintaining a path between each other. The route discovery typically consists of the network-wide flooding of a request message. To reduce overhead, the search area may be reduced by a number of optimizations [10. 25, 31].

The benefit of this approach is that signaling overhead is likely to be reduced compared to proactive approaches, particularly in networks with low to moderate traffic loads. When the number of data sessions in the network becomes high, then the overhead generated by the route discoveries approaches, and may even surpass, that of the proactive approaches. The drawback to reactive approaches is the introduction of a *route acquisition latency*. That is, when a route is needed by a source node, there is some finite latency while the route is discovered. In contrast, with a proactive approach, routes are typically available the moment they are needed. Hence, there is no delay to begin the data session.

10.3.1 Ad Hoc On-Demand Distance Vector Routing

The Ad Hoc On-Demand Distance Vector (AODV) Routing Protocol [48] provides on-demand route discovery in mobile ad hoc networks. Like most reactive routing protocols, route finding is based on a route discovery cycle involving a broadcast network search and a unicast reply containing discovered paths. Similar to DSDV, AODV relies on per-node sequence numbers for loop freedom and for ensuring selection of the most recent routing path. AODV nodes maintain a route table in which next-hop routing information for destination nodes is stored. Each routing table entry has an associated lifetime value. If a route is not utilized within the lifetime period, the route is expired. Otherwise, each time the route is used, the lifetime period is updated so that the route is not prematurely deleted.

When a source node has data packets to send to some destination, it first checks its route table to determine whether it already has a route to the destination. If such a route exists, it can use that route for data packet transmissions. Otherwise, it must initiate a route discovery procedure to find a route. To start route discovery, the source node creates a *route request* (RREQ) packet. It places in this packet the destination node's IP address, the last known sequence number for that destination, and the source's IP address and current sequence number. The RREQ also contains a hop count, initialized to zero, and a RREQ ID. The RREQ ID is a per-node, monotonically increasing counter that is incremented each time the node initiates a new RREQ. In this way, the source IP address, together with the RREQ ID, uniquely identifies a RREQ and can be used to detect duplicates. After creating this message, the source broadcasts the RREQ to its neighbors.

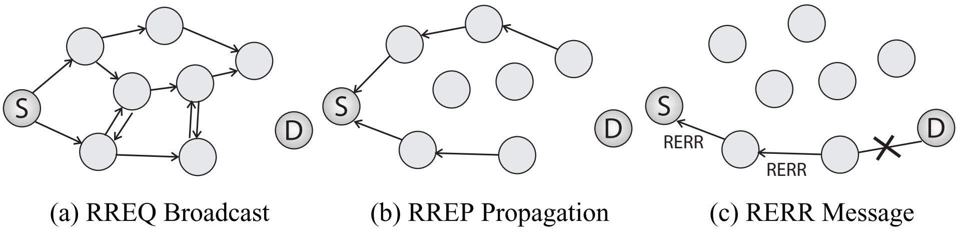
When a neighboring node receives a RREQ, it first creates a *reverse route* to the source node. The node from which it received the RREQ is the next hop to the source node, and the hop count in the RREQ is incremented by one to get the hop distance from the source. The node then checks whether it has an unexpired route to the destination. If it does not have a valid route to the destination, it simply rebroadcasts the RREQ, with the incremented hop count value, to its neighbors. In this manner, the RREQ floods the network in search of a route to the destination. Figure 10.2(a) illustrates this flooding procedure.

When a node receives a RREQ, it checks whether it has an unexpired route to the destination. If it does have such a route, then one other condition must hold for the node to generate a reply message indicating the route. The node's route table entry for the destination must have a corresponding sequence number that is at least as great as the indicated destination sequence number in the route request. That is,

$$dseq_{rt} \geq dseq_{RREQ}$$

When this condition holds, the node's route table entry for the destination is at least as recent as the source node's last known route to the destination. This condition ensures that the most recent route is selected, and also guarantees loop freedom (see [47] for a proof). Once this condition is met, the node can create a *route reply* (RREP) message. The RREP contains the source node's IP address, the destination node's IP address, and the destination's sequence number as given by the node's route table entry for the destination. In addition, the hop count field in the RREP is set equal to the node's distance from the destination. If the destination itself is creating the RREP, the hop count is set equal to zero. After creating the reply, the node unicasts the message to its next hop toward the source node. Thus, the reverse route that was created as the RREQ was forwarded is utilized to route the RREP back to the source node.

When the next hop receives the RREP, it first creates a *forward route* entry for the destination node. It uses the node from which it received the RREP as the next hop toward the destination. The hop count for that route is the hop count in the RREP, incremented by one. This forward route entry for the destination will be utilized if the source selects this path for data packet transmissions to the destination. Once the node creates the forward route entry,



(a) RREQ Broadcast (b) RREP Propagation (c) RERR Message

Figure 10.2. AODV route discovery and maintenance.

it forwards the RREP to the destination node. The RREP is thus forwarded hop by hop to the source node, as indicated in Figure 10.2(b). Once the source receives the RREP, it can utilize the path for the transmission of data packets. If the source receives more than one RREP, it selects the route with the greatest sequence number and smallest hop count.

Once a route is established, it must be maintained as long as it is needed. A route that has been recently utilized for the transmission of data packets is called an *active* route. Because of the mobility of the nodes, links along paths are likely to break. Breaks on links that are not being utilized for the transmission of data packets do not require any repair; however, breaks in active routes must be quickly repaired so that data packets are not dropped. When a link break along an active path occurs, the node upstream of the break (i.e., closer to the source node) invalidates the routes to each of those destinations in its route table. It then creates a *route error* (RERR) message. In this message it lists all of the destinations that are now unreachable due to the loss of the link. After creating the RERR message, it sends this message to its upstream neighbors that were also utilizing the link. These nodes, in turn, invalidate the broken routes and send their own RERR messages to their upstream neighbors that were utilizing the link. The RERR message thus traverses the reverse path to the source node, as illustrated in Figure 10.2(c). Once the source node receives the RERR, it can repair the route if the route is still needed.

AODV contains a number of optimizations and optional features [45]. To improve the protocol performance and reduce overhead, source nodes can utilize an *expanding ring search* to search for routes to the destination. The propagation of the RREQ is controlled by modifying the time to live (TTL) value of the packet. Incrementally larger areas of the network are searched until a route to the destination is discovered. If a route to the destination can be found in the local area, a network-wide flood can be avoided.

Another optimization is the local repair of link breaks in active routes. When a link break occurs, instead of sending a RERR to the source, the node upstream of the break can try to repair the link locally itself. If successful, fewer data packets are dropped because the route is repaired more quickly. If the local repair attempt is unsuccessful, a RERR message is sent to the source node as previously described.

In addition to these optimizations, AODV contains a number of optional features to improve operation in a wide range of scenarios. For instance, during route discovery, if only intermediate nodes respond and the destination never receives a copy of the RREQ, the destination will not necessarily have a route to the source node. If two-way conversation with the destination is desired, this lack of route from the destination to the source can be problematic. Hence, AODV defines a *gratuitous RREP* that can be sent to the destination node when a RREP is created by an intermediate node. This gratuitous RREP informs the destination of a route to the source, as if the destination had performed a route discovery. Another optional feature is the *RREP acknowledgment* (RREP-ACK). When unidirectional links are suspected, the RREP-ACK can be utilized to ensure the next hop received the RREP. If an RREP-ACK is not received, *blacklists* can be utilized to indicate unidirectional links so that these links are not used in future route discoveries. In addition, AODV allows the use of periodic Hello messages for monitoring connectivity to neighboring nodes.

10.3.2 Dynamic Source Routing

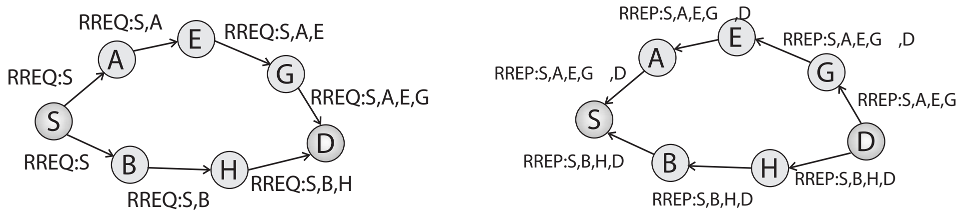
The Dynamic Source Routing (DSR) protocol [24] is similar to AODV in that it is a reactive routing protocol with a route discovery cycle for route finding. However, it has a few important differences.

One of the primary characteristics of DSR is that it is a source routing protocol; instead of being forwarded hop by hop, data packets contain strict source routes that specify each node along the path to the destination. Route request (RREQ) and route reply (RREP) packets accumulate source routes so that once a route is discovered, the source learns the entire source route and can place that route into subsequent data packets. Figure 10.3(a) illustrates the process of route discovery. The source node places the destination IP address, as well as its own IP address, into the RREQ and then broadcasts the message to its neighbors. When the neighboring nodes receive the message, they update their route to the source and then append their own IP addresses to the RREQ. Thus, as the RREQ is forwarded throughput the network, the traversed path is accumulated in the message. When intermediate nodes receive the RREQ, they can create or update routing table entries for each of the nodes listed in the source route, not just the source node.

When a node with a route to the destination receives the RREQ, it responds by creating a RREP. If the node is the destination, it places the accumulated source route from the RREQ into the RREP. Otherwise, if the node is an intermediate node, it concatenates its source route to the destination to the accumulated route in the RREQ, and places this new route into the RREP. Hence, in either scenario the message contains the full route between the source and the destination. The source route in the RREP is reversed and the RREP is unicast to the source. Note that as intermediate nodes receive and process the RREP, they can create or update routing table entries to each of the nodes along the source route. Figure 10.3(b) illustrates the propagation of two RREPs back to the source. When a link break in an established path occurs, the node upstream of the break creates a route error (RERR) message and sends it to the source node.

Instead of maintaining a route table for tracking routing information, DSR utilizes a *route cache*. The cache allows multiple route entries to be maintained per destination, thereby enabling *multipath* routing, as will be discussed in Section 10.7.1. When one route to a destination breaks, the source can utilize alternate routes from the route cache, if they are available, to prevent another route discovery. Similarly, when a link break in a route occurs, the node upstream of the break can perform *route salvaging,* whereby it utilizes a different route from its route cache, if one is available, to repair the route. However, even when route salvaging is performed, a RERR message must still be sent to the source to inform it of the break.

Other characteristics that distinguish DSR from other reactive routing protocols include the fact that DSR's route cache entries need not have lifetimes. Once a route is placed in the route cache, it can remain there until it breaks. However, timeouts, capacity limits, and cache-replacement policies have been shown to improve DSR's performance



(a) RREQ Broadcast

(b) RREP Propagation

Figure 10.3. DSR route discovery.

[20]. Additionally, DSR nodes have the option of *promiscuous listening,* whereby nodes can receive and process data and control packets that are not addressed, at the MAC layer, to themselves. Through promiscuous listening, nodes can utilize the source routes carried in both DSR control messages and data packets to gratuitously learn routing information for other network destinations. Finally, to reduce the overhead of carrying source routes in data packets, DSR also allows flow state to be established in intermediate nodes. This flow state effectively allows hop-by-hop forwarding with the same source-based route control as provided by the source route [21].

For a detailed study of the performance of AODV and DSR, as well as an explanation of how the protocol differences result in performance differentials, the reader is referred to [16].

10.4 GEOGRAPHICAL APPROACHES

Geographical approaches build on the proactive or reactive techniques previously described and in addition incorporate geographical information to aid in routing [3, 25, 33, 58, 59]. This geographical information can be in the form of actual geographic coordinates [as obtained through the global positioning system (GPS)], or can be obtained through reference points on some fixed coordinate system. The use of geolocation information can prevent network-wide searches for destinations, as either control packets or data packets can be sent in the general direction of the destination if the recent geographical coordinates for that destination are known. This reduces the control overhead generated in the network; however, all nodes must have continual access to their geographical coordinates for these approaches to be useful. The following describes one such geographical routing approach.

10.4.1 Location-Aided Routing

The Location-Aided Routing (LAR) protocol [25] is a reactive routing protocol that utilizes geographical coordinates to direct route request messages to the previously known location of the destination. The protocol defines two areas: the *expected zone* and the *request zone*. The expected zone is the area in which the destination is most likely to be discovered. To calculate this area, the source must know a previous location of the destination at time t_0, as well as an estimate of the velocity, v, at which the destination was traveling at t_0. If the current time is t_1, the expected zone can be calculated as a circle of radius $v(t_1 - t_0)$ centered at D. The request zone is the area in which the route request for the destination should propagate. In order to have the greatest probability of finding the destination, the request zone is defined to be the smallest rectangle that contains both the expected zone and the source node. Figure 10.4 illustrates an example expected and request zone.

The basic route discovery procedure of LAR is similar to that of other reactive routing protocols. When a source needs a route to a destination, it creates a route request (RREQ) message for that destination. If the source recently had a route to the destination, then the source calculates the expected zone and the request zone, and places the coordinates of the request zone boundary into the RREQ message. If the source does not have any previous information about the destination, then it is unable to calculate the expected and request zones. In this case, the algorithm defaults to basic flooding.

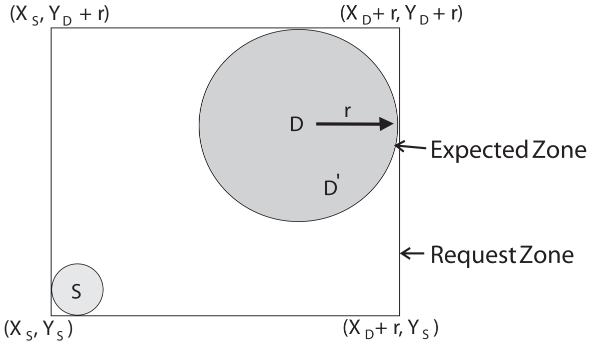


Figure 10.4. LAR expected and request zone.

When a node receives the RREQ, it processes it as described in the previous sections with the following exception. The node first determines whether it lies in the request zone defined in the RREQ. Because every node knows its current geographical coordinates, it can easily make this determination. If the node does not lie within the request zone, then it does not process the packet. Otherwise, if it does lie within the request zone, it processes the packet and either rebroadcasts it or sends a reply, depending on whether it has a current route to the destination.

The size of the request zone is a trade-off between control overhead and probability of finding the destination. A small request zone runs the risk of not including the area in which the destination is currently located. It is also possible that, although the destination may lie within the request zone, the path between the source and the destination may not be completely contained within this zone. In this case, a route to the destination will not be discovered. On the other hand, if the request zone is too large, the control overhead savings will be minimal.

In addition to the previously described approach, LAR also defines a second method for determining the request zone. Instead of calculating a rectangular area, the source places its distance from the previous location of the destination, along with the coordinates of the destination's previous location, in the RREQ. When neighboring nodes receive the RREQ, they calculate their distance from the destination ($DIST_i$) and then compare that value with the source's distance as reported in the RREQ ($DIST_S$). For some parameter δ, if $DIST_S + \delta \geq DIST_i$, then the node i processes the request. When it forwards the request, the node replaces $DIST_S$ in the RREQ with its distance $DIST_i$. On the other hand, if $DIST_S + \delta < DIST_i$, the node discards the RREQ. In practical implementations, δ typically equals 0. By using this approach, the nodes are forcing the RREQ to make forward progress to the estimate of the destination's location.

In both approaches, the RREQ is prevented from flooding the entire network because it is restricted to areas that are likely to be en route to the destination. This results in a reduction in both bandwidth and processing overhead.

10.5 HYBRID APPROACHES

The characteristics of proactive and reactive routing protocols can be integrated in various ways to form *hybrid* networking protocols. Hybrid networking protocols may exhibit proactive behavior given a certain set of circumstances, while exhibiting reactive behavior given a different set of circumstances. These protocols allow for flexibility based on the characteristics of the network. Hybrid approaches include the Zone Routing Protocol [41] and the Distance Routing Effect Algorithm for Mobility [3].

10.5.1 Zone Routing Protocol

The Zone Routing Protocol (ZRP) [41] integrates both proactive and reactive routing components into a single protocol. Around each node, ZRP defines a *zone* whose radius is measured in terms of hops. Each node utilizes proactive routing within its zone and reactive routing outside of its zone. Hence, a given node knows the identity of and a route to all nodes within its zone. When the node has data packets for a particular destination, it checks its routing table for a route. If the destination lies within the zone, a route will exist in the route table. Otherwise, if the destination is not within the zone, a search to find a route to that destination is needed. Figure 10.5 illustrates the zone concept. In this figure, the zone radius is two hops.

For intrazone routing, ZRP defines the Intrazone Routing Protocol (IARP). IARP is a link-state protocol that maintains up-to-date information about all nodes within the zone. For any given node *X*, *X*'s *peripheral nodes* are defined to be those nodes whose minimum distance to *X* is the zone radius. In Figure 10.5, *S*'s peripheral nodes are nodes *A*, *B*, *C*, and *D*. These peripheral nodes are important for reactive route discovery. ZRP utilizes the Interzone Routing Protocol (IERP) for discovering routes to destinations outside of the zone. For route discovery, the notion of *bordercasting* is introduced. Once a source node determines the destination is not within its zone, the source *bordercasts* a query message to its peripheral nodes. During the bordercast, the query message is relayed toward these peripheral nodes using trees constructed within the intrazone topology. After receiving the message, the peripheral nodes, in turn, check whether the destination lies within their zone. If the destination is not located, the peripheral nodes in turn bordercast the query message to their peripheral nodes. This process continues until either the destination is located, or until the entire network is searched. Once a node discovers the destination, it unicasts a reply message to the source node.

Figure 10.6 illustrates an example of the bordercast discovery procedure. In the figure, node *S* performs a query for the destination *X*. By using the IARP, it learns that *X* is not within its zone. It bordercasts the query message to its peripheral nodes. In the figure, the

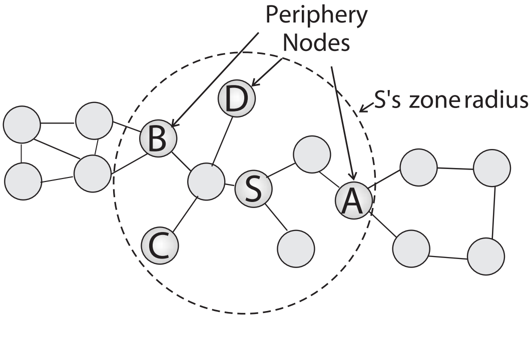


Figure 10.5. ZRP zone radius.

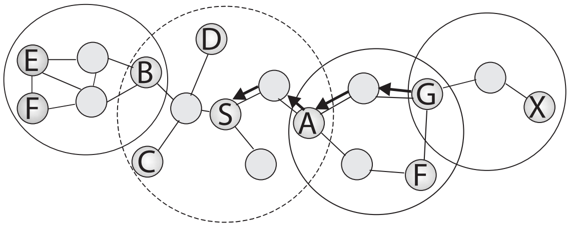


Figure 10.6. Example ZRP route discovery.

dotted circle represents the radius of *S*'s zone. The peripheral nodes, in turn, check their zone, and after not finding the destination, bordercast the query message to their peripheral nodes. The solid circles in the figure represent the forward propagation of the query messages to each node's peripheral nodes (i.e., the circles do not enclose nodes that have already received the query). Hence, only the portion of each node's zone that have not been previously traversed by the query message is shown. Eventually, node *G* discovers *X* within its zone, and then unicasts a reply back to node *S*.

To improve query efficiency, a random query processing delay can be used as an effective query control mechanism. By waiting a random interval between query reception and query forwarding, the chance of collisions during forwarding is reduced and, therefore, the effectiveness of the protocol is improved. In addition, ZRP defines other optimizations to reduce the messaging and processing overhead [19]. In particular, these include early termination of queries by preventing a query from propagating into a zone that has already been searched for a destination.

Recently, a new version of ZRP, ZRPv2, has been introduced [54]. ZRPv2 differs from the original ZRP in the manner in which bordercasting is performed. In both versions, route discovery is initiated with the query source node constructing a bordercast tree to its *uncovered* peripheral nodes. An *uncovered* node is one that does not belong to the routing zone of a node that already has received the query. The node then forwards the query message to its bordercast tree neighbors. When these neighbors receive the query message, rather than forwarding the message to the query source's downstream peripheral nodes (as in the original ZRP), they each construct bordercast trees to their own uncovered peripheral nodes, and forward the route query to their bordercast tree neighbors. Each node that receives a route query follows the above procedure until the destination, or a node possessing a fresh route to the destination, is reached. At that point a route reply is unicast back to the source.

Performing bordercasting on a hop-by-hop basis yields a uniform and, thus, simpler, protocol implementation. Also, the need for maintaining an extended routing zone is eliminated.

10.6 CLUSTERING AND HIERARCHICAL ROUTING

IP addresses are hierarchical in that they identify the location of the end device within the global Internet. Routing protocols in the Internet take advantage of this hierarchy when determining routes between networks. Ad hoc networks, on the other hand, are not necessarily able to take advantage of a hierarchy based on IP addresses. Nodes joining an ad hoc network are likely to have pre-assigned IP addresses from other networks. Hence, the nodes form a hodgepodge of addresses, and hierarchical routing based on addresses is not necessarily possible.

However, flat routing, as performed by the previously discussed protocols, has a number of disadvantages. The primary disadvantage is that it is not scalable. In the worst case, a node must maintain a routing table entry for every other node in the network, and the amount of information exchanged to create these routing table entries is $O(n^2)$, where n is the number of nodes in the network.

To increase the scalability of the ad hoc network, hierarchical, or clustering, protocols can be utilized. Hierarchical protocols place nodes into groups, often called clusters. These groupings may be based on a number of criteria, but most commonly they are based on either location [1, 2, 4, 5] or functionality [43, 60].

There have been numerous hierarchical routing protocols developed that take a variety of approaches toward clustering. In this section, a survey of the characteristics and algorithms for clustering is given; individual protocols are not examined.

The physical properties of the clusters vary between clustering protocols. As shown in Figure 10.7, clusters can be either overlapping or completely disjoint. Further, a one-level hierarchy can be created, or recursive multilevel hierarchies are also possible. Finally, control within a cluster can be held by a cluster leader, or the cluster can be completely distributed with no cluster leader. In networks in which cluster leaders exist, these leaders typically process control packets on behalf of their member nodes. It is also possible for cluster leaders to form a routing backbone within the network [4, 12]. This can be a desirable property if the cluster leaders are preselected as nodes with greater resources than nonleader nodes.

Figure 10.7 illustrates an example cluster topology with cluster leaders. The cluster boundaries are based on the transmission range of the cluster leaders. All nodes within a cluster must be within direct transmission range of the cluster leader. In this example, cluster boundaries are allowed to overlap. Nodes that are located within the boundaries of multiple clusters are called *gateways*. These nodes serve as routers between the two clusters. With many clustering protocols, it is also possible for pairs of nodes to serve as *distributed* or *joint* gateways. Distributed gateways are a pair of nodes that are within direct transmission range of each other, where each node lies in a different cluster. Together, the two nodes can be used to route between clusters.

Cluster leaders are typically initialized through some distibuted algorithm. For instance, there can be a leader election algorithm, in which the node with the highest ID

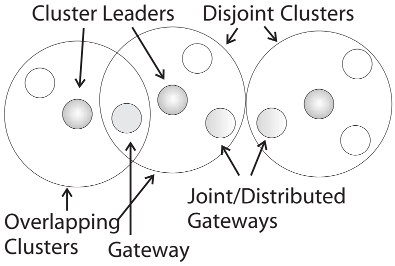


Figure 10.7. Example cluster configuration.

within some area becomes the leader for that area [12]. Alternatively, weights, such as number of neighbors, transmission range, and so on, can be used instead of the ID of the node [2, 7]. Other algorithms take a more "first come, first elected" approach [4]. In this method, when a node joins a network, it queries its one-hop neighbors to determine whether it is within range of an existing cluster leader. If so, the node may choose to join the already existing cluster. Otherwise, if there is not a nearby cluster leader, the node becomes a leader itself. Using this approach, during the initialization of a network, the first node to join the network would become a cluster leader.

Once the network has been initialized and cluster leaders have been established, there must be leader selection and revocation algorithms in place. The leader selection algorithm specifies under what conditions a node should become a cluster leader. For instance, if a node wanders to the periphery of the network, it may lose contact with all current cluster leaders. In this case, it may become a cluster leader itself. Similarly, the network must also have in place a cluster revocation algorithm. Without such an algorithm, there will be a slow, continual growth in the number of leaders as more nodes wander to the network perimeter and become leaders. In the worst case, without a revocation algorithm, it would be possible for each network node to eventually become a leader. If all nodes became leaders, the hierarchy would become ineffective.

There are a number of leader revocation algorithms. Most commonly, when two leaders come within direct transmission range of each other, one of the leaders must give up its leader status. The leader to give up its status may be the leader with the lowest ID [12], or a weight-based approach may be used [2, 7]. An alternate method has been proposed in [4], where, instead of requiring one node to become a nonleader whenever two leaders come within transmission range, a leader gives up its leader status only when its cluster becomes a *subset* of the other cluster. This approach has some beneficial properties such as preventing a rippling effect, where one leadership change results in further network leadership changes. Because leadership changes are expensive in terms of control overhead and routing changes, they should be minimized overall.

There are two key benefits of utilizing clustering protocols in an ad hoc network. The first of these is that they enable hierarchical routing. Consider the network in Figure 10.8. Suppose a source node *S* wants to send packets to the destination *D*. The path it might discover with one of the previously discussed flat routing protocols is

$$S \rightarrow C_1 \rightarrow G_1 \rightarrow C_2 \rightarrow D$$

Using a flat routing scheme, whenever any link along this path breaks, the route must be repaired with a route maintenance procedure (i.e., sending a route error to the source, local repair, route salvaging, etc.). However, with a hierarchical routing scheme, the path is instead recorded as

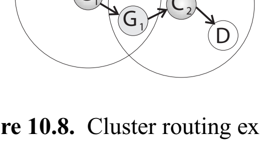


Figure 10.8. Cluster routing example.

$$S \rightarrow C_1 \rightarrow C_2 \rightarrow D$$

The difference is that, with the hierarchical route, the path is recorded at the cluster level. Hence, the route is recorded from cluster leader C_1 to C_2, and the intermediate node is not specified. To get from C_1 to C_2, *any* gateway node connecting the two clusters can be utilized. For instance, gateway G_1 may be selected to route between the clusters. In the event that G_1 wanders out of transmission range of one of the clusters and can no longer serve as a gateway, one of the other gateways (i.e., G_2) can be used instead. Because of this increased routing flexibility, link breaks do not necessarily result in route repairs if another gateway node is available. Decreasing the number of route repairs decreases the amount of control overhead generated in the network, and, consequently, increases the number of data packets that can be delivered to the destination.

The second benefit of hierarchical protocols is that the hierarchy can be used to implement hierarchical addressing schemes. Addresses can be assigned to nodes based on their cluster membership. For instance, in the network shown in Figure 10.9, a node *z* is a member of a cluster *y*. In that case, its address may be *y.z*. If the hierarchy consisted of multiple levels, then cluster *y* might in turn be within a larger cluster *x*. In this case, the node's address would be *x.y.z*. If the node were a member of multiple clusters, the node could have multiple, concurrent addresses. In mobile networks, multiple addresses would be beneficial because at any given time, at least one of the addresses is likely to still be valid.

As was seen in the previous example, multilevel hierarchies can be created. Multilevel hierarchies that dynamically adjust their depth based on the network topology can further increase the scalability of the network. In multilevel hierarchies, each cluster becomes a node at the next-highest cluster level. Routes can be recorded up and down the hierarhical routing tree. A route ascends the hierarchical tree only as high as needed to reach the branch containing the destination. Multilevel hierarchical routing protocols include those in [5, 28, 44, 52, 56].

Hierarchical routing protocols have many clear advantages. They improve route robustness by increasing routing flexibility; routes that are recorded between clusters, as opposed to between nodes, have more routing options, and, hence, can be repaired more easily. Increasing route robustness leads to an increase in route lifetimes, thereby resulting in fewer route reconstructions, less control traffic from route repairs, and increased data delivery. However, there are also disadvantages that many hierarchical routing protocols suffer from. To create and maintain the clusters, many clustering protocols require periodic overhead [1, 4, 64]. Such overhead is needed to maintain current information about cluster memberships and gateway availability. To overcome this drawback, some clustering

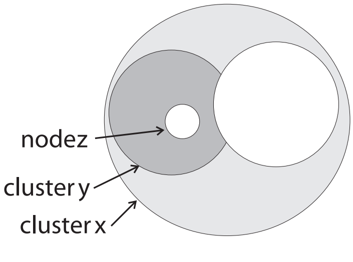


Figure 10.9. Hierarchical addressing.

approaches have opted for a more on-demand approach, such that clusters are only created when needed [18]. This can eliminate a significant fraction of the overhead in networks with low traffic demands. Other disadvantages include the centralization of routes through cluster leaders. If the cluster leaders can be selected solely from nodes with greater resources, than this centralization is likely to be beneficial. However, in networks without specialized, high-power nodes, centralizing routes unfairly taxes leader nodes, and is likely to result in premature depletion of their batteries. Protocols that utilize cluster leaders for the establishment of routes but that do not actually require cluster leaders to participate on the routes can eliminate this unfair burden. Finally, the cluster leader–gateway–cluster leader routing requirement can result in the use of nonshortest paths.

10.7 OTHER TECHNIQUES

10.7.1 Multipath Routing

The majority of routing protocols previously described utilize route tables for storing routing information. In such tables, there is one next-hop entry for each destination. Although in the IP route table it is only possible to store one route entry for each destination, it is possible in the routing protocol to create a route cache, in which multiple routing entries per destination can be maintained. In the event that the path in use to some destination breaks or becomes otherwise unusable, an alternate route in the route cache can be utilized instead. When alternate routes are available, route discoveries can be saved, and control overhead can be reduced.

There are numerous proposals for multipath routing. The vast majority of these study multipath routing with on-demand routing approaches. The DSR protocol utilizes route caches for maintaining multiple paths to each destination. In [20], various caching strategies and cache configurations are analyzed.

Other proposals investigate modifications to existing protocols to add multipath routing capability. For instance, the Ad Hoc On-Demand Multipath Distance Vector (AOMDV) routing protocol is described in [36]. In AOMDV, multiple routing entries are stored per destination, and there is one lifetime value associated with all the routing entries for a destination. Hence, all the routing possibilities are refreshed or expire at the same time. Also, the emphasis in AOMDV is on finding multiple routes that are link disjoint, implying that the routes do not share any common links.

Another approach is taken in AODV-BR [29], which proposes back-up paths for routing around link breaks in active routes. When a link break in an active route occurs, the node closer to the source sends an error message to the source, and also broadcasts the data packet to its neighbors. The packet indicates in the data header that the link has broken and that the packet is in need of alternate routing. When the neighboring nodes receive the packet, they forward it to the destination if they have a route for that destination.

There are additional proposals that describe new protocols to handle multipath routing [30, 39, 50]. One of these, the Split Multipath Routing (SMR) protocol [30], operates similarly to the reactive protocols described in Section 10.3; however, the protocol makes the following modifications. To obtain maximal node disjoint paths, intermediate nodes are not allowed to respond to route requests. Further, route requests and route replies contain entire routing paths. Intermediate nodes forward the first route request they receive, and only rebroadcast subsequent ones if the requests arrive from different neighbors and have not traveled further from the source than the first route request received. When the destination receives the request, it responds to the first request, and then waits a time interval to receive additional requests. At the end of the time interval, it responds to the request that traveled the route most disjoint from the original request received. The protocol can be easily configured so that the destination responds to more than two requests.

Finally, [42] investigates the impact of alternate path routing on ad hoc routing protocols, particularly focusing on load balancing and the concurrent utilization of multiple routes. In particular, it studies the impact of *route coupling* in ad hoc networks with only a single available channel. Route coupling occurs when two routes have nodes or links in common. The authors discover that, while alternate routes enable a reduction in end-to-end delay, the route coupling results in an underutilization of network resources, thereby preventing alternate path routing from achieving significant performance improvements.

10.7.2 Energy-Conserving Protocols

All of the routing protocols previously described in this chapter were designed with the characteristics of a mobile environment in mind. Hence, each protocol attempts to reduce control overhead and processing requirements so as to minimize power utilization. However, there is a set of ad hoc routing protocols that have been designed with the specific goal of further minimizing energy consumption. These protocols take a variety of approaches toward energy efficiency, including powering down unutilized nodes, load balancing, and dynamic transmission power adjustment.

The Geographical Adaptive Fidelity (GAF) algorithm [61] is an example of an energy-conservaing protocol that powers down unutilized nodes. GAF is able to identify *routing-equivalent* nodes in a network so that unnecessary nodes can be turned off. GAF nodes utilize location information, such as that provided through GPS, to create a virtual grid. All nodes within a given grid square are equivalent with respect to routing functionality. Nodes in the grid square can therefore coordinate to determine the sleep duration for each node. The nodes can be coordinated such that load balancing aids in the overall energy utilization of each node; each node takes a turn forwarding data packets.

Routing based on overall energy cost and remaining battery lifetimes is performed by the Battery Energy Efficient (BEE) protocol [13]. This protocol assigns to each route a cost function that takes into account both energy cost and battery lifetime. When a source initiates a data stream, it evaluates the cost function for all the possible loop-free routes to the destination. It selects the route that minimizes the cost function. Similar to this work, [11] describes an approach that balances path selection and energy consumption rates among the network nodes in proportion to the available energy reserves of the nodes.

A number of protocols dynamically adapt node transmission ranges to reduce overall power consumption. For instance, the approach described in [17] computes the power cost along the discovered paths from the source to the destination. The path with the minimum such cost is selected. The network nodes use a power management scheme such that nodes are grouped into clusters, and each node transmits at the minimum power level to reach each of the nodes in the cluster. Along these same lines, the protocol proposed in [51] dynamically adjusts node transmit powers to maintain a connected topology using minimum power. Nodes in dense portions of the network decrease their transmission power to reach fewer nodes, whereas nodes in remote areas of the network increase their transmission power to become more fully connected. It should be noted, however, that increasing the number of hops in a path has the drawback of increasing the likelihood of a link break in that path. This can result in an increase in control overhead due to the additional route repairs.

In addition to routing protocols, there is a wide range of research on power-efficient MAC protocols and transport layer protocols. For instance, the Power-Aware Multiple Access Protocol with Signalling (PAMAS) [57] utilizes information gleaned from RTS/CTS exchanges to turn off nodes when they are not receiving packets. Because the RTS and CTS messages contain the length of an upcoming data packet, nodes receiving this message that are not the sender or receiver of the data packet can safely turn off their interfaces while the data packet is being transmitted.

10.7.3 Security-Aware Protocols

The open nature of wireless communication and the portability of mobile devices creates a challenge for securing mobile networks. In contrast to wired networks, intruders do not need to compromise network hosts to gain access to the network; malicious nodes need only be within transmission range of a node in order to participate in and overhear network traffic. Although securing ad hoc networks is a challenge, it is a necessary component for ad hoc networks to be universally deployed. Recent ad hoc network security research has addressed a wide range of security topics, from secure routing to intrusion detection and monitoring. In this section, a sampling of approaches from these two areas is highlighted.

10.7.3.1 Secure Routing. When performing an on-demand route discovery such as those described in Section 10.3, there are a number of possible attacks. Malicious nodes can lie about the existence of paths and can modify the information in routing messages to influence path selection. Further, a source node has no method for determining whether a path actually exists when it receives a route reply. However, in nearly every application scenario a user would like to be assured that his or her data traffic is actually reaching its intended destination. To address this issue, a number of *secure routing protocols* have been developed. Some of these protocols attempt to secure existing routing protocols, such as those described in Section 10.3. Others are new routing protocols that have been designed with the primary goal of security.

An example of this latter type of protocol is the Authenticated Routing for Ad Hoc Networks (ARAN) Protocol [55]. ARAN is based on certificates and assumes that all network nodes can obtain a certificate from a trusted certificate server before joining the ad hoc network. The certificate contains the public key of the node. ARAN utilizes a route discovery procedure similar to AODV. A source node *S* generates a Route Discovery Packet (RDP) for a destination. The RDP is signed with the source's private key and contains its certificate. When a neighbor *A* receives the RDP message, it verifies the signature of the source by extracting *S*'s public key from its certificate, and then sets up a reverse path back to the source node. The node then signs the contents of the message, appends its own certificate, and broadcasts the message to its neighbors. When *A*'s neighbor *B* receives the message, it validates *A*'s signature, and then replaces this signature with its own signature (the signature of the source node is retained). The packet continues to be rebroadcast in this manner until it reaches the destination. When the first RDP reaches the destination, the destination node verifies the signature of the source node and then sends a digitally signed route reply packet (REP) back to the source. The REP travels the same path as the RDP, and the same signing procedure is performed by intermediate nodes. Because the destination must sign the REP message, only the destination is allowed to respond to the

RDP. Also, because RDP messages are signed at each hop and do not contain a hop count or a source route, malicious nodes have no opportunity to intentionally redirect traffic.

The Secure Routing Protocol (SRP) [40] is another approach to secure routing that is based on the assumption of the existence of a security association between the source and destination node. To initiate communication, the two nodes negotiate a shared secret key. A message authentication code (MAC) is used to ensure that the reply message is not modified en route to the source node. Only the destination can respond to route query messages, and the source is assured that the destination was reached because the shared key is used as input to the MAC computation.

Like SRP, Ariadne assumes that all pairs of communicating nodes have secret MAC keys [22]. Each pair of nodes maintains two keys, one for each direction of communication. Ariadne is utilizes symmetric cryptographic primitives and is based on the DSR routing protocol and the TESLA broadcast authentication protocol [49]. Ariadne has the properties that source and destination nodes can authenticate each other due to the secret keys, and that the source node can authenticate each entry on the path returned in the route reply. In addition, a one-way function is utilized so that no intermediate node can remove a previous node in the source route contained in route request and reply messages.

The Security Aware Routing (SAR) protocol described in [62] relies on trust levels to provide security. Nodes form a trust hierarchy whereby each node is assigned a specific trust level. Designed to run over a reactive routing protocol such as AODV or DSR, route request and reply messages are assigned a security level by the source node. Only nodes with at least the indicated level of security can process and forward the control messages. Hence, SAR discovers routes in which all nodes along the path meet the desired level of security.

Finally, a mechanism for securing the AODV protocol is presented in [63]. The protocol, SAODV, utilizes digital signatures and hash chains for securing AODV control messages. The digital signatures are utilized to authenticate the nonmutable fields of the control messages, whereas the hash chains are used to secure the hop count information. The approach assumes the nodes have access to a key management system so that the nodes can obtain the public keys of the other nodes within the network.

10.7.3.2 Intrusion Detection and Monitoring Schemes. Intrusion detection is a mechanism widely used in wired networks to detect malicious invaders and trigger an appropriate response. Intrusion detection in ad hoc networks is somewhat less straightforward because membership in the network is open to virtually any user. Hence, it is difficult to detect when a user is actually a malicious intruder. Nevertheless, intrusion detection techniques can be employed in ad hoc networks to detect misbehaving nodes, particularly in networks in which the membership is well defined. Such networks include military networks and collaborative networks comprised of a team of individuals.

An intrusion detection and response mechanism is presented in [65]. It uses the cooperative statistical anomaly detection model to protect against attacks on routing protocols or other wireless applications and services. Each intrusion detection system (IDS) agent runs independently and monitors local activities. Intrusions are detected from local traces, and responses are subsequently initiated. If an anomaly is detected in the local data, or if the evidence is inconclusive and a broader search is warranted, neighboring IDS agents cooperate to participate in global intrusion detection actions.

Another approach taken by a handful of protocols is to monitor node behavior to detect misbehaving nodes. Node misbehavior can come in many forms; however, a common action to monitor is whether a node en route to a destination forwards data packets that it receives for that destination. For instance, the monitoring system described in [37] uses nodes called watchdogs to monitor the forwarding of data packets by intermediate nodes. After a node transmits a data packet, it *promiscuously* listens to determine whether the next hop along the path forwards the data packet to its next hop. For this functionality to work, a node must know the identity of the node two hops further along the path. In addition to the watchdog, network nodes also run a pathrater module to determine the reliability of paths. Nodes maintain ratings for other network nodes and, hence, select paths with the highest aggregate rating. The watchdog system is used to maintain the rating for neighboring nodes.

A similar monitoring system is described in [9]. This approach incorporates a trust manager and reputation system with a path manager to detect misbehaving or nonconforming nodes and exclude them from routing. The difference with this approach is that nodes propagate information about detected misbehaving nodes so that those nodes can be excluded from participation in the network. When a node receives such a message indicating the misbehavior of another node, the node must be able to authenticate the source of that message. This prevents denial of service attacks by malicious nodes against other, benign nodes.

10.8 CONCLUSION

As has been shown in this chapter, there exists a vast variety of routing protocols designed specifically for ad hoc mobile networks. These networks create a hostile routing environment due to the mobility of the nodes and the resulting ephemeral nature of the network links. However, significant strides have been made toward the development of robust routing protocols that can deliver high percentages of traffic, even in dynamic environments.

It is likely that there does not exist a single routing protocol that can solve the needs of every conceivable ad hoc network scenario. Rather, the selection of a routing protocol for a given network is likely to be dependent upon the dominating characteristics of that network. Hence, certain routing protocols are likely to perform best in networks of one set of characteristics, while others will perform better in networks with a differing set of characteristics. More work is needed to identify the sets of characteristics that promote the optimum behavior of each individual protocol and class of protocols.

ACKNOWLEDGMENTS

This work is largely based on a tutorial on mobile ad hoc networks created jointly by the author and Sung-Ju Lee of Hewlett-Packard Laboratories. The author would like to thank Thomas Clausen, Zygmunt Haas, Yih-Chun Hu, Sung-Ju Lee, Richard Ogier, Marc Pearlman, Prince Samar, and Fred Templin for their insightful comments and contributions to the protocol descriptions in this chapter.

REFERENCES

1. D. J. Baker and A. Ephremides. “The Architectural Organization of a Mobile Radio Network via a Distributed Algorithm,” *IEEE Transactions on Communications, 29*(11), 1694–1701, November 1981.

2. S. Basagni. “Distributed Clustering for Ad Hoc Networks,” in *Proceedings of the 1999 International Symposium on Parallel Architectures, Algorithms, and Networks,* pp. 310–315, June 1999.

3. S. Basagni, I. Chlamtac, V. Syrotiuk, and B. Woodward. “A Distance Routing Effect Algorithm for Mobility (DREAM),” in *Proceedings of the 4th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom),* pp. 76–84, Dallas, TX, October 1998.

4. E. M. Belding-Royer. “Hierarchical Routing in Ad hoc Mobile Networks.” *Wireless Communications and Mobile Computing,* 2002.

5. E. M. Belding-Royer. “Multi-Level Hierarchies for Scalable Ad Hoc Routing.” *Wireless Networks,* 2002.

6. B. Bellur, R. G. Ogier, and F. L. Templin. “Topology Broadcast Based on Reverse-Path Forwarding (TBRPF).” *IETF Internet Draft, draft-ietf-manet-tbrpf- 01.txt,* (work in progress), March 2001.

7. C. Bettstetter and R. Krausser. “Scenario-Based Stability Analysis of the Distributed Mobility-Adaptive Clustering (DMAC) Algorithm,” in *Proceedings of the 2nd Annual Symposium on Mobile Ad hoc Networking and Computing,* Long Beach, California, October 2001.

8. J. Broch, D. A. Maltz, D. Johnson, Y.-C. Hu, and J. Jetcheva. “A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols,” in *Proceedings of the 4th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom),* pp. 85–97, Dallas, Texas, October 1998.

9. S. Buchegger and J.-Y. L. Boudec. Nodes Bearing Grudges: “Towards Routing Security, Fairness, and Robustness in Mobile Ad Hoc Networks,” in *Proceedings of the Tenth Euromicro Workshop on Parallel, Distributed and Network-based Processing,* pp. 403–410, Canary Islands, Spain, IEEE Computer Society, January 2002.

10. R. Castaneda and S. R. Das. “Query Localization Techniques for On-demand Routing Protocols in Ad Hoc Networks,” in *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom),* pp. 186–194, Seattle, August 1999.

11. J.-H. Chang and L. Tassiulas. “Energy Conserving Routing in Wireless Ad-Hoc Networks,” in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM),* pp. 22–31, Tel Aviv, Israel, March 2000.

12. C.-C. Chiang, H.-K. Wu, W. Liu, and M. Gerla. “Routing in Clustered Multihop, Mobile Wireless Networks with Fading Channel,” in *Proceedings of IEEE Singapore International Conference on Networks (SICON),* pp. 197–211, April 1997.

13. C.-F. Chiasserini and R. R. Rao. “Routing Protocols to Maximize Battery Efficiency,” in *Proceedings of IEEE MILCOM,* Los Angeles, CA, October 2000.

14. T. Clausen, P. Jacquet, A. Laouiti, P. Muhlethaler, A. Qayyum, and L. Viennot. “Optimized Link State Routing Protocol,” in *Proceedings of IEEE INMIC,* Lahore, Pakistan, December 2001.

15. S. R. Das, R. Castaneda, and J. Yan. “Comparative Performance Evaluation of Routing Protocols for Mobile, Ad Hoc Networks,” in *Proceedings of the 7th International Conference on Computer Communications and Networks,* pp. 153–161, Lafayette, LA, October 1998.

16. S. R. Das, C. E. Perkins, and E.M. Royer. “Performance Comparison of Two Ondemand Routing Protocols for Ad Hoc Networks,” in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM),* pp. 3–12, Tel Aviv, Israel, March 2000.

17. T. A. ElBatt, S. V. Krishnamurthy, D. Connors, and S. Dao. “Power Management for Throughput Enhancement in Wireless Ad Hoc Networks,” in *Proceedings of the IEEE International Conference on Communications (ICC),* pp. 1503– 1513, New Orleans, LA, June 2000.

18. M. Gerla, T. Kwon, and G. Pei. “On Demand Routing in Large Ad Hoc Wireless Networks with Passive Clustering,” in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC),* Spetember 2000.

19. Z. J. Haas and M. R. Pearlman. “The Performance of Query Control Schemes for the Zone Routing Protocol.” *ACM/IEEE Transactions on Networking, 9*(4), 427– 438, August 2001.

20. Y.-C. Hu and D. B. Johnson. “Caching Strategies in On-Demand Routing Protocols for Wireless Ad Hoc Networks,” in *Proceedings of the Sixth Annual IEEE/ACM International Conference on Mobile Computing and Networking (MobiCom 2000),* pp. 231–242, Boston, MA, August 2000.

21. Y.-C. Hu and D. B. Johnson. “Implicit Source Routing in On-Demand Ad Hoc Network Routing,” in *Proceedings of the Second Symposium on Mobile Ad Hoc Networking and Computing* (MobiHoc 2001), pp.1–10, Oct. 2001.

22. Y.-C. Hu, D. B. Johnson, and A. Perrig. “Ariadne: A Secure On-Demand Routing Protocol for Ad Hoc Networks,” in *Proceedings of the 8th ACM International Conference on Mobile Computing and Networking (Mobicom),* Atlanta, GA, September 2002.

23. P. Johansson, T. Larsson, N. Hedman, B. Mielczarek, and M. Degermark. “Scenario-based Performance Analysis of Routing Protocols for Mobile Ad-Hoc Networks,” in *Proceedings of the 5th ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom),* pp. 195–206, Seattle, WA, August 1999.

24. D. B. Johnson and D. A. Maltz. “Dynamic Source Routing in Ad Hoc Wireless Networks,” in T. Imielinski and H. Korth (Eds.), *Mobile Computing,* pp. 153–181. Kluwer Academic Publishers, 1996.

25. Y.-B. Ko and N. H. Vaidya. “Location-Aided Routing (LAR) in Mobile Ad Hoc Networks,” in *Proceedings of the 4th ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom),* pp. 66–75, Dallas, Texas, October 1998.

26. R. Kravets and P. Krishnan. “Application-Driven PowerManagement for Mobile Communication.” *Wireless Networks, 6*(4), 263–277, 2000.

27. A. Laouiti, A. Qayyum, and L. Viennot. “Multipoint Relaying: An Efficient Technique for Flooding in Mobile Wireless Networks,” in *Proceedings of the 35th Annual Hawaii International Conference on System Sciences (HICSS’ 2002),* Waikoloa, HI, January 2002.

28. G. S. Lauer. “Packet-Radio Routing,” in M. Steenstrup (Ed.), *Routing in Communications Networks.* Prentice-Hall, 1995.

29. S.-J. Lee and M. Gerla. “AODV-BR: Backup Routing in Ad Hoc Networks,” in *Proceedings of the Wireless Communications and Networking Conference (WCNC),* Chicago, IL, September 2000.

30. S.-J. Lee and M. Gerla. “Split Multipath Routing with Maximally Disjoint Paths in Ad Hoc Networks,” in *Proceedings of the IEEE International Conference on Communications (ICC),* pp. 3201–3205, Helsinki, Finland, June 2001.

31. S.-J. Lee, E. M. Royer, and C. E. Perkins. “Ad Hoc Routing Protocol Scalability.” *International Journal on Network Management,* 2002.

32. S.-J. Lee, C.-K. Toh, and M. Gerla. “A Simulation Study of Table-Driven and On-Demand Routing Protocols for Mobile Ad-Hoc Networks.” *IEEE Network, 13*(4), 48–54, July/August 1999.

33. J. Li, J. Jannotti, D. S. J. D. Couto, D. R. Karger, and R. Morris. “A Scalable Location Service for Geographic Ad hoc Routing,” in *Proceedings of the 6th ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom),* pp. 120–130, Boston, MA, August 2000.

34. J. Macker and M. S. Corson. Internet Engineering Task Force (IETF) Mobile Ad Hoc Networks (MANET) Working Group Charter. http://www.ietf.org/html.charters/manet-charter.html.

35. G. S. Malkin and M. E. Steenstrup. “Distance-Vector Routing,” in M. Steenstrup (Ed.), *Routing in Communications Networks,* pp. 83–98. Prentice-Hall, 1995.

36. M. Marina and S. Das. “On-demand Multipath Distance Vector Routing in Ad Hoc Networks,” in *Proceedings of the International Conference on Network Protocols (ICNP),* Riverside, CA, November 2001.

37. S. Marti, T. J. Giuli, K. Lai, and M. Baker. “Mitigating Routing Misbehavior in Mobile Ad Hoc Networks,” in *Proceedings of the 6th Annual ACM/IEEE International Conference on Mobile Computing and Networking,* pp. 255–265, 2000.

38. J. Moy. “Link-State Routing,” in M. Steenstrup (Ed.), *Routing in Communications Networks,* pp. 135–157. Prentice-Hall, 1995.

39. A. Nasipuri and S. Das. “On-DemandMultipath Routing for Mobilc Ad Hoc Networks,” in *Proceedings of the IEEE Conference on Computer Communications and Networks (ICCCN),* pp. 64–70, Boston, MA, October 1999.

40. P. Papadimitratos and Z. Haas. “Secure Routing for Mobile Ad Hoc Networks,” in *Proceedings of the SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS 2002),* San Antonio, Texas, January 2000.

41. M. R. Pearlman and Z. J. Haas. “Determining the Optimal Configuration for the Zone Routing Protocol.” *IEEE Journal on Selected Areas in Communications, 17*(8), 1395–1414, August 1999.

42. M. R. Pearlman, Z. J. Haas, P. Sholander, and S. S. Tabrizi. “On the Impact of Alternate Path Routing for Load Balancing in Mobile Ad Hoc Networks,” in *Proceedings of the 1st Annual Workshop on Mobile and Ad hoc Networking and Computer (MobiHOC)),* pp. 3–10, Boston, August 2000.

43. G. Pei, M. Gerla, and X. Hong. LANMAR: “Landmark Routing for Large Scale Wireless Ad Hoc Networks with Group Mobility,” in *Proceedings of the 1st Annual Workshop on Mobile and Ad hoc Networking and Computer (MobiHOC),* pp. 11–18, Boston, MA, August 2000.

44. G. Pei, M. Gerla, X. Hong, and C.-C. Chiang. “A Wireless Hierarchical Routing Protocol with Group Mobility,” in *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC),* pp. 1538–1542, New Orleans, September 1999.

45. C. E. Perkins, E. M. Belding-Royer, and S. R. Das. “Ad Hoc On-Demand Distance Vector (AODV) Routing.” *IETF Internet Draft, draft-ietf-manet-aodv-10.txt,* March 2002. (Work in Progress).

46. C. E. Perkins and P. Bhagwat. “Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers,” *SIGCOMM ’94: Computer Communications Review, 24*(4), 234–244, October 1994.

47. C. E. Perkins and E. M. Royer. “Ad-Hoc On-Demand Distance Vector Routing,” in *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications,* pp. 90–100, New Orleans, February 1999.

48. C. E. Perkins and E. M. Royer. “The Ad Hoc On-Demand Distance Vector Protocol,” in C. E. Perkins (Ed.), *Ad Hoc Networking,* pp. 173–219. Addison-Wesley, 2000.

49. A. Perrig, R. Canetti, D. Song, and J. Tygar. “Efficient and Secure Source Authentication for Multicast,” in *Proceedings of the Network and Distributed System Security Symposium (NDSS),* San Diego, February 2001.

50. J. Raju and J. Garcia-Luna-Aceves. “A New Approach to On-Demand Loop- Free Multipath Routing,” in *Proceedings of the IEEE Conference on Computer Communications and Networks (ICCCN),* pp. 522–527, Boston, October 1999.

51. R. Ramanathan and R. Rosales-Hain. “Topology Control of Multihop Wireless Networks Using Transmit Power Adjustment,” in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM),* pp. 404–413, Tel Aviv, Israel, March 2000.

52. R. Ramanathan and M. Steenstrup. “Hierarchically-organized,Multihop Mobile Wireless Networks for Quality-of-Service Support.” *ACM/Baltzer Mobile Networks and Applications, 3*(1), 101–118, 1998.

53. E. M. Royer and C.-K. Toh. “A Review of Current Routing Protocols for Ad-Hoc Mobile Networks.” *IEEE Personal Communications, 6*(2), 46–55, April 1999.

54. P. Samar, M. R. Pearlman, and Z. J. Haas. “Hybrid Routing: The Pursuit of an Adaptable and

Scalable Routing Framework for Ad Hoc Networks," in M. Ilyas (Ed.), *Handbook of Ad Hoc Wireless Networks,* Chapter 14. CRC Press, 2002.

55. K. Sanzgiri, B. Dahill, B. N. Levine, C. Shields, and E. M. Belding-Royer. "A Secure Routing Protocol for Ad Hoc Networks," in *Proceedings of the 10th IEEE International Conference on Network Protocols (ICNP),* Paris, November 2002.
56. N. Shacham. "Hierarchical Routing in Large, Dynamic Ground Radio Networks," in *Proceedings of the 18th Hawaii InternationalConference on System Sciences,* pp. 292–301, 1985.
57. S. Singh, M. Woo, and C. S. Raghavendra. "Power-Aware Routing in Mobile Ad hoc Networks," in *Proceedings of the 4th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom),* pp. 181–190, Dallas, Texas, October 1998.
58. I. Stojmenovic. "Location Updates for Efficient Routing in Ad Hoc Wireless Networks," in *Handbook of Wireless Networks and Mobile Computing,* pp. 451–471. Wiley, 2002.
59. I. Stojmenovic. "Position Based Routing in Ad hoc Mobile Networks." *IEEE Communications Magazine, 40*(7), 128–134, July 2002.
60. P. F. Tsuchiya. "The Landmark Hierarchy: A New Hierarchy for Routing in Very Large Networks," in *Computer Communication Review,* pp. 35–42, Stanford, CA, August 1988.
61. Y. Xu, J. Heidemann, and D. Estrin. "Geography-informed Energy Conservation for Ad Hoc Routing," in *Proceedings of the 7th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom),* pp. 70–84, Rome, July 2001.
62. S. Yi, P. Naldurg, and R. Kravets. "A Security Aware Routing Protocol forWireless Ad Hoc Networks," in *Proceedings of the 6th World Multi-Conference on Systemics, Cybernetics and Informatics (SCI),* pp. 286–292, Orlando, FL, July 2002.
63. M. G. Zapata and N. Asokan. "Securing Ad Hoc Routing Protocols," in *Proceedings of the ACM Workshop onWireless Security (WiSe),* Atlanta, GA, September 2002.
64. J. Zavgren. "NTDR Mobility Management Protocols and Procedures," in *Proceedings of the IEEE Military Communications Conference (MILCOM),* pp. 292–301, November 1997.
65. Y. Zhang and W. Lee. "Intrusion Detection in Wireless Ad hoc Networks," in *Proceedings of the 6th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom),* pp. 26–33, Seattle, August 1999.

CHAPTER 11

ENERGY-EFFICIENT COMMUNICATION IN AD HOC WIRELESS NETWORKS

LAURA MARIE FEENEY

11.1 INTRODUCTION

One reason why working on energy-efficient communication in ad hoc networks is so much fun is the complexity of trade-offs available to the designer of energy-aware systems. The richness of interactions among the physical elements of the system, the various layers of the protocol stack, and the environment in which the system operates requires creative and careful attention to obtain interesting and meaningful results.

This chapter surveys current work on energy-efficient communication in ad hoc wireless networks, focusing on problems and approaches that are most specific to the decentralized ad hoc environment and illustrate most clearly its unique challenges. In addition, I have chosen to emphasize practical issues and approaches and to focus on work that is largely based on readily available hardware and communication technology.

The chapter opens with a brief introduction to ad hoc wireless networks and discusses characteristics that make these networks structurally different from infrastructure wireless networks, necessitating the development of new energy management techniques.

The second section motivates the goal of energy-efficient communication by presenting some results obtained by measuring the energy consumption of various (mostly IEEE 802.11-based) devices. One key result is that the energy consumed by an idle network interface dominates total energy consumption.

The third section examines some existing and proposed power save protocols. Power save protocols attack the problem of high energy consumption in an idle network interface by selecting intervals during which the interface can use a low energy consumption sleep state, with minimal impact on overall network performance.

Mobile Ad Hoc Networking. Edited by Basagni, Conti, Giordano, and Stojmenovic.
ISBN 0-471-37313-3 © 2004 Institute of Electrical and Electronics Engineers, Inc.

The fourth section presents the topology control and minimum energy routing problems. In a multihop network, nodes can alter their transmit power level to manipulate the effective network topology, reducing interference and increasing network capacity, as well as reducing energy consumption.

The fifth section discusses maximum lifetime routing. In an ad hoc network, nodes cooperate to forward traffic from a source to a destination. A node that forwards traffic on behalf of other nodes may exhaust its energy reserves, so that it can no longer participate in the network. It is therefore necessary to select paths in a way that maximizes the network lifetime.

The appropriate metric for network lifetime depends to some extent on the application scenario. In a sensor network, data is forwarded to distinguished gateway nodes, which are the only destinations in the network. Such a network is usually modeled as a dense, uniform collection of functionally equivalent nodes: As long as sensing and communication coverage is maintained, the lifetime of any individual sensor is relatively unimportant. In a personal communication network, nodes are devices associated with individuals. This means that any node can be a destination and that loss of connectivity to any node is significant.

There are two recurring themes in this chapter. The first is that problems of energy efficiency cannot be isolated to a single layer in the protocol stack. The second is the extent to which wireless propagation and energy consumption models affect the design and evaluation of energy-efficient techniques. Examples appear throughout the chapter, which closes with some discussion of these themes.

11.2 AD HOC WIRELESS NETWORKS

In an ad hoc wireless network, nodes cooperatively form a network independently of any fixed base station infrastructure. These networks are generally characterized by bandwidth-constrained, variable-capacity links and an unpredictable, dynamic topology. Each node communicates directly with destinations within wireless transmission range and indirectly with all other destinations, relying on its peers to forward traffic on its behalf. Ad hoc routing is an active area of research [31] and we now find a complete alphabet soup of proposed routing protocols, ranging from AODV [28] to ZRP [15]. The nodes of an ad hoc network also cooperate to provide application services such as service discovery, namespace and session management, and security [14].

Because the nodes of an ad hoc network are usually small, battery powered devices, energy management is a critical issue for practical deployment of these networks.[1] Ad hoc networks differ from wireless infrastructure networks in two fundamental ways that require unique strategies to obtain energy-efficient behavior.

First, infrastructure wireless networks have a strongly asymmetric structure. Although the mobile units are battery constrained, the base stations have no such limitation. Low-level energy management strategies are often based on spending energy at the base station to conserve energy at the mobile node. This is not a viable approach in an ad hoc network, which has no such fixed elements.

Second, in an infrastructure wireless network, nodes operate independently of each other, using the base station to communicate with other nodes and access services in the

[1] Vehicle-based ad hoc networks are a significant exception.

infrastructure network. Energy management strategies therefore only need to consider the node and its local applications. In an ad hoc network, nodes are highly interdependent and must cooperate to provide routing and other services, making it important to maximize the network lifetime. Greedy strategies, in which each node seeks only to minimize its own energy consumption, are not effective in this context.

11.3 MEASURING ENERGY CONSUMPTION

Before considering various approaches to reducing communication energy consumption, it is reasonable to question whether the network interface contributes significantly to the overall energy consumption of a mobile system.

The variety of devices, operating modes, energy management techniques, and usage scenarios make it impossible to make blanket statements about energy consumption in portable devices. Obviously, measurements of specific systems quickly become outdated. Nevertheless, measurements [35] show that the network interface represents a significant fraction of the energy consumed by a laptop PC and is dominant source of energy consumption in some PDA hardware. More recently, in a preliminary implementation of a Bluetooth-based sensor device [12, 18], the interface accounted for over 40% of the total energy consumption when the Bluetooth device was in standby mode. Moreover, the relative cost of communication may be expected to increase as advances continue to be made in low-power hardware and energy-efficient operating systems and applications. This trend will accelerate as communication functionality is increasingly incorporated into small, specialized devices such as sensors.

The design and evaluation of energy-efficient communication protocols therefore requires practical understanding of the energy consumption behavior of the underlying network interface. The energy consumed by an interface depends on its operating mode: In the sleep state, an interface can neither transmit nor receive, so it consumes very little energy. To be able to transmit or receive, an interface must explicitly transition to the idle state, which requires both time and energy. In the idle state, an interface can transmit or receive data at any time, but it consumes more energy than it does in the sleep state, due to the number of circuit elements that must be powered.

Because of their wide availability, low-cost and relatively stable, open specification, IEEE 802.11-based [16] interfaces have attracted considerable attention. Table 11.1 summarizes some experimental measurements of the power consumption of various network interfaces. Although the data vary somewhat among the various manufacturers, models, and measurement methods, there are consistent patterns. Transmitting requires more ener-

Table 11.1. Some Power Consumption Measurements

Interface	Transmit	Receive	Idle	Sleep	Mbps
IEEE 802.11 Interfaces (2.4 GHz)					
Aironet PC4800 [8]	1.4–1.9 W	1.3–1.4 W	1.34 W	0.075 W	11
Lucent Bronze [10]	1.3 W	0.97 W	0.84 W	0.066 W	2
Lucent Silver [10]	1.3 W	0.90 W	0.74 W	0.048 W	11
Cabletron Roamabout [6]	1.4 W	1.0 W	0.83 W	0.13 W	2
Lucent WaveLAN [20]	3.10 W	1.52 W	1.5 W	—	—

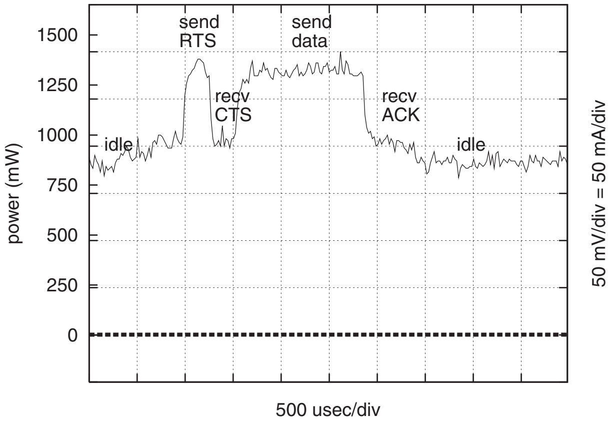


Figure 11.1. Unicast transmission—256 bytes at 2 Mbps (Lucent IEEE 802.11 card).

gy than receiving, but the difference is much less than a factor of two. The idle energy consumption is quite high, comparable to that of receiving and an order of magnitude more than that of sleeping.

More detailed results are presented in [10, 8], which describe direct measurements of a network interface card as it transmits and receives packets of varying sizes, as in Figure 11.1. In [10], this data is used to develop a packet-level energy consumption model for a Lucent IEEE 802.11 network interface. The energy consumed (in addition to the baseline idle energy consumption) when the interface transmits, receives, or discards a packet can be described using, at each operation, an incremental component that is proportional to the size of the packet and a fixed component that reflects channel acquisition and other overhead. (The effects of contention and retransmissions are not addressed in this work.)

Although the specific numerical results are of limited value, again there are consistent patterns. The fixed overhead is high, due to the cost of the RTS/CTS/data/ACK handshake and the size of the IEEE 802.11 MAC headers. The fixed overhead dominates for packets smaller than 338 bytes (2 Mbps) or 1.2 Kbytes (11 Mbps).[2] Nevertheless, it is the energy consumption of the idle state that dominates total energy consumption. A rough calculation based on [10] shows that an IEEE 802.11 interface sending 10 128-byte broadcasts (2 Mbps) per second and receiving the same from each of four neighbors consumes only about 1% more power than an idle interface.

The results obtained from such experiments can be incorporated into packet-level simulations of ad hoc routing protocols, as described in [9]. It is clear that bandwidth and energy are not analogous metrics: minimizing bandwidth usage does not necessarily minimize energy consumption. In particular, the results show that in a moderately dense network, broadcast traffic can be expensive, due to the multiplied cost of receiving. Promiscuous mode operation is similarly costly.

The experiments described in [10] did not measure the effects of transmit-power control on the energy consumption at the network interface. Measurements of the Aironet PC4800B card, which supports multiple transmit-power levels, are reported in [8]. The re-

[2] The difference is due to the fact that IEEE 802.11 control traffic is transmitted at the (slower) base data rate.

sults show that as the the output power level decreases from 50 mW to 1 mW, the power consumed by the transmitter decreases about 500 mW. This variation represents only about 25% of the total power consumption; the baseline power consumption accounts for about 70% of the total. The power consumption of the receiver is comparable to the baseline and roughly independent of the transmit-power level.

The conclusion to be drawn from these data is obvious. To reduce the energy consumption of the network interface, it is necessary to find a way for the interface to spend more time in the sleep state and less time awake in the idle state. Such power-save protocols are the topic of the next section.

11.4 POWER-SAVE PROTOCOLS

A power-save protocol puts a node's network interface into the sleep[3] state in order to save energy. A sleeping node cannot forward or receive traffic and its unavailability may interrupt the flow of traffic though a multihop ad hoc network. Power-save protocols therefore seek to maximize energy saving, while minimizing impact on throughput, latency, and route latency. Two main classes of power-save protocols, network layer protocols and MAC layer protocols, are introduced below.

11.4.1 Network-Layer Power-Save Protocols

Network-layer protocols make up the largest class of power-save protocols. Scheduling of the interface is driven by network-layer traffic, which is buffered at the MAC layer for sleeping neighbors or routed so as to take advantage of nonsleeping ones. Power-save protocols are based on three basic strategies, outlined below:

1. The first and simplest approach is a synchronized power-save mechanism. Nodes periodically wake up to listen to announcements of pending traffic, and remain awake to exchange traffic, if necessary. The restricted windows for announcing and forwarding traffic can result in high latencies, however. Establishing the required phase synchronization can be also problematic in a dynamic, multihop ad hoc network.
2. A second approach is based on network topology. Some covering set of the network is defined such that it is topologically representative of the network. The nodes in the covering set provide connectivity equivalent to that of the full network, so that the remaining nodes can spend most of their time in the sleep state with minimal impact on network performance. Such protocols can be either synchronous or asynchronous, both in determining the covering set and in traffic forwarding.
3. A third approach is intended for fully asynchronous operation, in which nodes maintain independent and possibly even dissimilar sleep–wake schedules. The scheduling rules are defined such that neighboring nodes' schedules are guaranteed to eventually overlap. Retransmission rules are defined such that a bounded number of attempts are required to permit two nodes to establish connectivity.

[3] A node whose network interface is in the idle or sleep state is referred as an idle or sleeping node, without implying anything about the state of components such as the CPU. Energy-aware joint scheduling of computing, storage, and communication subsystems is an area of active research.

11.4.2 Synchronous Power-Save Protocols

In a synchronous power-save protocol, nodes periodically wake up and exchange traffic announcements and pending traffic. One difficult aspect of these protocols is determining intervals and announcement windows that maximize energy savings, while minimizing impact on throughput and latency. This approach also requires that nodes maintain a globally synchronized sleep–wake cycle, meaning that they must share arbitrary phase information.

The power-save protocol defined in the IEEE 802.11 standard is a synchronous power-save protocol. Each node explicitly associates itself with a BSS base station (basic service set) or an IBSS (independent basic service set) and synchronizes itself with its established beacon interval. This method works well for scenarios in which a single network and source of phase synchronization can be conveniently designated. This requirement is unfortunate because the ad hoc model is specifically designed to support flexible methods of establishing a network. Ad hoc networks are intended for deployment independent of any centralized administration and should expect to experience arbitrary partitions and merges.

Note that the problem of phase synchronization is not the same as the problem of clock synchronization, which can be achieved using specialized hardware, such as GPS receivers. Consider, for example, the case of two nearby wireless clouds, which were formed separately and therefore have differently phased sleep–wake cycles, despite having the same period. In order for the two clouds to merge, they must first discover each other's existence; then they must either synchronize their sleep–wake cycles or select nodes to translate between the two networks. The problem is complicated in a multihop network, in which two nodes might simultaneously initiate independent merges. Although not insoluble, fully addressing this problem can involve considerable complexity and overhead, especially in the case of large dynamic networks.

11.4.3.1 Power Saving in IEEE 802.11. IEEE 802.11 IBSS power save is particularly relevant to the ad hoc model, although there are differences between a multihop wireless network and an IBSS network, in which each node explicitly discovers and associates itself with a single, connected IBSS.

In IEEE 802.11 power save, a synchronized beacon interval is established by the node that initiates the IBSS and is maintained in a distributed fashion. The IBSS also defines a common fixed-length, ad hoc traffic indication message (ATIM) window, which occurs at the beginning of each beacon interval. All nodes in the IBSS wake up at the beginning of the beacon interval and remain awake until the end of the ATIM window. At the beginning of the beacon interval, nodes randomly contend to transmit the synchronization beacon, synchronizing themselves with the first beacon they receive. Each node then transmits an ATIM to every other node for which it has pending unicast traffic. Each node that receives an ATIM responds with an ATIM acknowledgment. Announcements of broadcast and multicast traffic are sent to the appropriate broadcast or multicast address, but are not acknowledged. Only beacons, announcements, and acknowledgments are transmitted during the ATIM window, avoiding contention with data traffic. At the end of the ATIM window, nodes that have not sent or received ATIM announcements go back to sleep. All other nodes remain awake throughout the remainder of the beacon interval, in order to send and receive the announced traffic.

Obviously, the effectiveness of the power-saving mechanism depends on the values selected for the beacon and ATIM intervals, as well as on the offered load. If the ATIM window is too short, not enough traffic is announced during the ATIM window to fully utilize the beacon interval. If the ATIM window is too long, not only are nodes required to spend a greater part of each interval awake, but more traffic may be announced than can be sent in the remainder of the beacon interval. Similarly, if the beacon interval is too short, the overhead of the sleep–wake cycle, beaconing, and traffic announcements will be high. If the beacon interval is too long, more nodes will announce traffic at each ATIM window and more destinations will remain awake after the ATIM window. Contention will also increase, due to the increased number of nodes trying to transmit in each interval.

Although power-saving mechanisms are part of the IEEE 802.11 standard, there appear to be relatively few published research results that examine their effectiveness. Two of these, [41] and [6], are discussed below. Unfortunately, the results are only moderately encouraging.

Energy consumption is the main criterion in evaluating a power-save protocol, but factors such as latency, throughput, and distribution of power consumption must also be taken into account. Time spent in the sleep state is only an indication of the actual energy savings, which will be reduced by the costs of the state transition, beaconing, and ATIM traffic, all of which are sensitive to these configuration parameters. Analysis of these protocols therefore depends on measurements such as those described in the previous section, as well as on traffic and mobility models.

Simulations described in [41] studied the effectiveness of the IEEE 802.11 power-save protocol for a fully connected eight-node IBSS. The experiment measured throughput and time spent in the sleep state for a variety of beacon intervals, ATIM window lengths, and offered loads. The results show considerable dependence on the beacon interval: Short beacon intervals give superior energy saving, but at the cost of substantially reduced throughput. As a general observation, the authors suggest that “if we were to sacrifice about 10% in throughput, we could save up to 30% energy.” However, such savings are obtained only at quite moderate loads; as offered load increases from 15% to 30%, the available savings declines substantially.

The choice of ATIM interval also depends on the offered load. For a moderately to heavily loaded network and a wide range of beacon intervals, throughput is maximized when the ATIM window occupies about 25% of the beacon interval. Because a smaller window results in lower energy consumption while still providing acceptable throughput for lightly loaded networks, the authors recommend adopting an adaptive ATIM window.

Simulations described in [6] were intended to assess the performance of the authors’ proposed power-save protocol, operating in conjunction with IEEE 802.11 power save. The results indicate that, by itself, IEEE 802.11 power save does not significantly reduce energy consumption in a multihop ad hoc environment. The authors suggest that this partly reflects poor handling of broadcast traffic, which forces all the nodes to remain awake through the beacon interval in which the broadcast is announced. The geographic routing protocol in these simulations used periodic broadcast, making this weakness particularly apparent. The results also show that for a multihop network, the cumulative latencies become extremely high because each packet is forwarded at most one hop in each beacon interval.

11.4.3 Topology-Based Power-Save Protocols

Topology-based power-save protocols are based on selecting a subset of nodes that are topologically representative of the full network. The nodes in this covering set remain in the idle state and are responsible for forwarding traffic in the network. Other nodes spend most of their time sleeping, waking up periodically to participate in subset election or to receive pending traffic. Topology-based protocols must therefore provide specific mechanisms for selecting the covering set and for forwarding traffic in the partial network defined by the covering set.

The problem of selecting an optimal covering set is nontrivial, especially as it must be a low-overhead, localized computation. The covering set must be chosen so as to maintain the effective capacity of the network and minimize the impact of the power-save protocol on throughput and latency. It must be recomputed in response to node failure and mobility. In addition, because nodes in the covering set are in the idle state rather than the sleep state, they consume much more energy. Care must therefore be taken to ensure that this role is rotated among nodes in the network, so as to maximize the network lifetime.

Although topology-based protocols are not inherently synchronized, they may rely on synchronized mechanisms for buffering traffic for sleeping nodes. In addition, many topology-based power-save protocols use connectivity information to determine the covering set. In this case, nodes that are not currently part of the covering set must exchange traffic to determine their connectivity. Some scheduling mechanism is needed to support this process, because it cannot be mediated by the covering set. It is also possible to select a covering set indirectly or probabilistically. Such techniques lend themselves to asynchronous operation.

The following subsections present three approaches to topology-based power saving: One is an algorithm for calculating dominating sets, whereas the other two, Span and GAF, are fully defined protocols.

11.4.3.1 Dominating Sets. A dominating set of a network is a subset of nodes, such that each node is either in the dominating set or is a neighbor of a node in the dominating set. In a connected dominating set, the dominating nodes form a connected subgraph of the network, as in Figure 11.2. The routing backbone that is formed by a connected dominating set is an obvious choice for use in topology-based power-save protocols.

Finding a minimal size dominating set is known to be computationally hard. Ad hoc networking protocols based on dominating sets must therefore define some distributed algorithm for approximating a dominating set. Most selection algorithms use a two-phase approach, such as that evaluated in [42]. This approach generally requires some synchronization to ensure consistent operation. In the first phase, nodes exchange neighbor information, and any node that has two unconnected neighbors marks itself as part of the con-

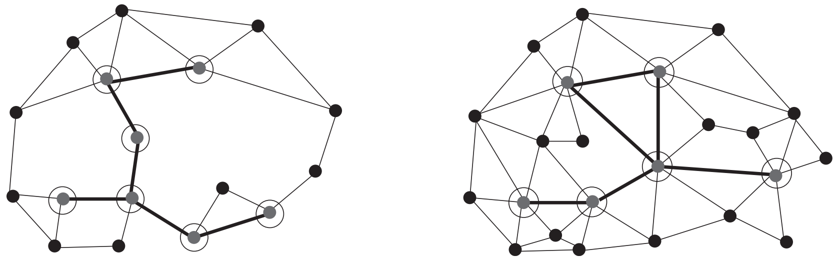


Figure 11.2. Connected dominating sets: the relative size of the dominating set (circled nodes) depends on network density.

nected dominating set. This set may be much larger than the minimum one. The second phase therefore eliminates any marked node that is redundant because its one-hop neighborhood is contained within the one-hop neighborhood of an adjacent marked node or is contained within the union of two such one-hop neighborhoods. Whenever possible, the node with the lowest energy reserves or with the lowest node degree is preferentially removed from the dominating set.

Connected dominating sets found using this algorithm can be used as a routing backbone for the network, although [42] does not present a protocol for dominator selection or packet forwarding. This means that a packet-level simulation cannot used be used to evaluate its performance. The simulation experiments instead evaluate analytically the proportion of nodes selected for the dominating set and the resulting network lifetime, but cannot fully address issues such as contention, throughput, latency, or packet loss.

In this work, the power consumption of dominator and nondominator nodes was modeled in an interesting way. Rather than simply specifying fixed values, the incremental cost of acting as a dominating node over some interval is specified as a function of the the routing overhead, which is proportional to the number of nodes in the network, and the forwarding overhead, which is inversely proportional to the number of forwarders. The relative magnitude of these factors depends on the average node degree, path length, and path lifetime (i.e., mobility), as well as the relative costs of transmitting and receiving.

Simulation results show that after the elimination phase, 30–40% of the nodes are in the dominating set. The elimination rule protecting nodes with low energy reserves, in particular, significantly extended the network lifetime. In the case of low mobility and low routing overhead, however, simple clustering also performed quite well: Although a high proportion of network nodes were designated as high-energy-consumption clusterheads, the forwarding load was also divided among the many clusterheads, resulting in good overall performance. This result closely reflects the modeling of forwarding overhead in the energy consumption model. It suggests that in some cases, a smaller dominating set is not necessarily better, even though these nodes have higher energy consumption. It is not clear whether similar results would be seen in a packet-level simulation, which would include factors such as routing overhead and contention among the larger number of forwarding nodes.

The use of a topology-dependent energy consumption model also exhibited another interesting effect. For all values of routing overhead and for all mobility levels, network lifetime *decreased* asymptotically as the node density increased. A fixed energy consumption model, on the other hand, yields a roughly linear *increase* in network lifetime as a function of density. This increase was also observed in packet-level simulation[4] of several other power-save protocols (discussed below). Simulation results for Span show network lifetime increasing roughly linearly with network density, whereas GAF shows a strong linear relationship between network lifetime and node density. This intriguing difference highlights the impact that the energy consumption model can have on simulation results.

11.4.3.2 Span. Span [6] is a fully specified power-save protocol, based on a routing backbone that is a connected dominating set, whose members are called "coordinators." Coordinators are continually in the idle state, whereas noncoordinator nodes wake up periodically to exchange traffic with the coordinator nodes and participate in coordinator election. The coordinators act as a low-latency routing backbone for the network and

[4] Note that node densities in the packet-level simulations were higher (20–80 vs. 4–20 nodes in transmit range).

buffer traffic for sleeping destinations, in effect acting as base stations for the noncoordinator nodes.

The coordinator election algorithm is structurally similar to the one described above, in that nodes provisionally join the dominating set, then eliminate themselves from it. Nodes periodically exchange HELLO messages to discover their two-hop neighborhood. A node marks itself eligible to be a coordinator if it discovers that two neighbors cannot communicate directly or via other coordinators. Each marked node schedules a backoff interval, during which it listens for announcements from other nodes. If the node is still eligible after this interval (i.e., no other suitable coordinators have announced themselves), it sends its own coordinator announcement. The backoff interval has both random and adaptive elements. Nodes with greater utility, that is, effectiveness at connecting new pairs of neighbors, and higher energy reserves announce themselves as coordinators more quickly than less effective ones, which volunteer later and only if they are still needed to complete the connected dominating set. After spending some time as a coordinator, a node withdraws as a coordinator, allowing other nodes to consider their eligibility and announce themselves as coordinators. Rotating the coordinator role in this way tends to balance nodes' energy reserves, even in the case of initially unequal distribution.

The coordinators buffer traffic for their sleeping neighbors, using the traffic announcement mechanism of IEEE 802.11. Because coordinators do not sleep, they have no need for the traffic announcement mechanism and a portion of each beacon interval is therefore reserved for traffic between coordinators. The routing protocol is integrated with the coordinator mechanism so that data is forwarded through the coordinator backbone with low latency until it is buffered by the appropriate coordinator for delivery to a sleeping destination.

Packet-level simulations using ns-2 [37] suggest that, compared to IEEE 802.11 power saving, Span provides about 50% energy saving in dense (12–78 nodes/transmit area) networks, with only minimal impact on throughput and packet loss. The savings increase only slightly with node density, due to the increasing overhead of the traffic announcement mechanism. There is also a two- to four-fold increase in latency, which also increases with node density. Rotation of the coordinator role equalizes energy consumption and the network lifetime increases 50–100%, as discussed above. The results also corroborate the experimental energy measurements. Even when Span is used to limit idle energy consumption, sending and receiving traffic accounts for well under 10% of the total energy consumed.

Span is a synchronous power-save protocol for two reasons. First, nodes must be awake simultaneously to exchange traffic to determine their connectivity and participate in coordinator election—the topology cannot be determined solely by the coordinators. Second, the underlying buffering and traffic announcement mechanism is based on the synchronous IEEE 802.11 power-save mechanism. This is not integral to Span operation, however; some form of asynchronous polling is a possible alternative.

11.4.3.3 GAF. Geographic adaptive fidelity (GAF) [45] is a power-save protocol that selects its representative nodes based on position information rather than membership in a dominating set. As defined, GAF is primarily intended for sensor networking scenarios. Nodes that are data sources or sinks do not participate in the power-save protocol, and there is no concept of buffering pending traffic for a sleeping node.

GAF partitions the network using a geographic grid. The grid size is defined such that each node in a grid square is within transmission range of every node in each adjacent grid square, implying a grid size of $R/\sqrt{5}$, where R is the node transmission range. This grid structure ensures that all the nodes in a grid square are equivalent with respect to providing connectivity to any adjacent grid square. One nonsleeping node in each grid square is sufficient to maintain the connectivity of the original network.

Because connectivity is defined by the grid, selecting the active node for each grid square does not require explicit exchange of connectivity information. Each node transitions independently among three states: sleep, discovery, and active. Nodes periodically wake up from the sleep state and transition to the discovery state. In the discovery state, a node listens for other nodes' announcements and can announce its own grid position ID and residual energy status. If the node hears no "higher ranking" announcement, it transitions to the active state, otherwise it transitions back to the sleep state. A node in the active state is responsible for maintaining network connectivity on behalf of its grid square, periodically announcing its state. After spending some time in the active state, a node transitions back to the discovery state, allowing the active role to be rotated among the nodes in the grid square.

The ranking function and state timeouts can be used to tune GAF, trading energy consumption against the risk that there will be no active node in a grid square. The ranking function is used to balance energy consumption among nodes, by preferring nodes with the longest "expected node active time," which is based on the node's residual energy and the length of time it is projected to remain in its current grid square. The sleep intervals are calculated such that nodes are likely to transition from the sleep state to the discovery state in time to replace an active node, if needed.

Currently, the ad hoc routing protocol operates independently of GAF. This makes it possible to isolate the impact of GAF power saving on routing-protocol performance. It imposes some burden on GAF, because when the active node in a grid square changes, the routing protocol interprets this as a route failure from which it must recover. Alternatively, GAF might be more closely coupled with an ad hoc routing protocol by using preemptive route recovery or grid-based forwarding.

The effect of GAF on the energy consumption and performance of AODV routing protocol has been studied in ns-2 packet-level simulation. In general, AODV and AODV/GAF have similar data delivery ratios and transfer delays, whereas the mean energy consumption per node is reduced by 40–50%. As discussed earlier, increasing node density provides proportional increases in network lifetime (defined as 80% data delivery ratio). Similar results were obtained using DSR [17] as the routing protocol.

One must be cautious, however, when dealing with geographic protocols, especially when position information is not used only for directional forwarding. In general, it is not possible to rely on a strong deterministic relationship between position and connectivity. Buildings, foliage, and terrain all have a dramatic effect on path loss, even between nodes which are geographically close together. In a real network, there may exist grids in which no single node (and possibly even no combination of nodes) has connectivity with every node in an adjacent grid.

The appropriate selection of the grid size helps to ensure the expected connectivity; an overly conservative choice reduces energy savings. Within a grid, the announcement mechanism can adjust somewhat to actual connectivity. If connectivity is poor, some nodes in the grid square may not receive an announcement. More than one node may choose to take on the role of an active node in the grid square, providing increased robustness, as well as increased energy consumption. Moreover, even if there are grids that are disconnected from one or more adjacent grids, the routing protocol may be able to route around these "holes" and maintain connectivity between source and destination. This is clearly an issue that emphasizes the importance of having good terrain and propagation models for use in evaluating such protocols.

11.4.4 Asynchronous Power-Save Protocols

Asynchronous power-save protocols allow nodes to maintain independent, possibly even dissimilar, schedules. The scheduling rules are defined such that neighboring nodes' awake intervals eventually overlap and retransmission rules are defined such that neighbors can eventually exchange traffic. Because initiating communication at each hop can take some time, this approach has the disadvantage of introducing latency, with broadcast traffic and route latency being particularly affected.

11.4.4.1 BECA/AFECA. The BECA/AFECA protocol [44, 43], a predecessor of GAF, does not rely on position information. Like GAF, it is intended for use in sensor networks. It is designed to be integrated with an on-demand ad hoc routing protocol, which discovers routes by means of broadcast flooding. BECA/AFECA operates asynchronously, maintaining network connectivity through strong timing dependencies with the associated routing protocol.

In the basic energy conservation algorithm (BECA), each node independently transitions between the sleep state and one of two logical idle states: listening and active. In the absence of traffic, a node alternates between the sleep state and the listening state. Once a node transmits or receives traffic, it transitions to the active state. Nodes in the active state return to the sleep state only after they have been not received traffic for some time.

Because nodes make these transitions independently, there is no guarantee that there is a live path to the destination, or even that the destination itself is awake at the time a route request (RREQ) is generated. In order to ensure that a RREQ reaches its destination, there must be a careful relationship among the various timing parameters of the power-save protocol and the route discovery protocol.

The fundamental unit is the listening interval, which is matched to the RREQ retry interval for the routing protocol. If the sleep interval is some integral multiple k of the listening/retry interval, then it will take at most $k + 1$ broadcasts for every neighbor to receive the RREQ. Once a node receives the RREQ, it transitions to the active state. As long as the timeout for the active interval is greater than the RREQ retry interval, the intermediate nodes will remain awake until the route discovery process has completed, taking at most $D(k + 1)$ route discoveries, where D is the path length.

Once the route has been established, only the nodes that forward traffic will remain active. The other nodes will not forward traffic, and once their active intervals have timed out, they will return to the low-energy-consumption sleep–listen cycle. Once traffic along the route ceases, the nodes on the route also time out and return to sleep–listen cycle.

This probabilistic approach requires that broadcast flooding of the RREQ be repeated several times, with increasing redundancy each time. As long as route discovery and repair are relatively rare operations, this is a minor drawback. If not, or if the network supports other services that also rely on broadcast, then this overhead becomes more of an issue. Moreover, because "logical broadcast" occurs over a relatively long interval, there is a risk of synchronization problems for higher-layer protocols and applications.

The adaptive fidelity energy conservation algorithm (AFECA) is an extension of BECA in which nodes adapt their sleep interval depending on the estimated network density. The more nodes, the smaller the proportion that must be awake to achieve the connectivity needed to forward traffic.

The performance of BECA/AFECA has been studied using ns-2 simulation packet-level simulation, with AODV as the routing protocol. Overall energy saving was on the order of 35%–45%, across a range of traffic loads, with a minimum sleep interval of 10 seconds. There was a significant increase in route latency, which averaged well under one second for unmodified AODV, but averaged between six and ten seconds using AFECA. AFECA also exhibited slightly higher loss rates than unmodified AODV.

AFECA makes no attempt to rotate forwarding functionality among nodes or to otherwise balance energy consumption across the network. In fact, because nodes that have recently forwarded traffic remain awake, they are more likely to participate early in the route discovery process and may therefore be more likely to be designated as forwarding node for additional routes. Depending on traffic patterns, this feedback can lead to unequal distribution of routing load and poorly distributed energy consumption. This is reflected in the simulation results. The network half-life increases by as much as 50%, but there is almost no increase in time to first node failure.

11.4.5 MAC Layer Power-Save Protocols

CSMA MAC layer power-save protocols use information derived from the media-access control process to find intervals during which the network interface does not need to be awake. While a packet is being transmitted, nearby nodes whose transmissions might interfere with the ongoing transmission must remain silent. These nodes can sleep, with little or no impact on throughput. Figure 11.3 shows an example based on the IEEE 802.11 MAC.

11.4.5.1 PAMAS. The PAMAS [33] protocol is based on this principle. PAMAS uses an RTS/CTS-style mechanism with a separate control signaling channel. A node that is waiting to initiate a transmission or is in the process of receiving a transmission causes other nodes to defer their transmissions by generating a busy tone on the control channel.

A PAMAS node turns itself off if a neighbor is transmitting and it has no packets to transmit or if it has a packet to transmit, but a neighbor is receiving. The node can deter-

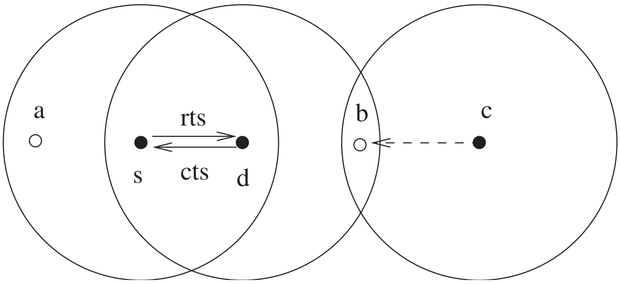


Figure 11.3. When *s* transmits to *d*, *a* hears the RTS and *b* hears the CTS, so *a* and *b* cannot transmit. (If *c* is outside the carrier sensing range of *s*, *c* can broadcast, with *b* as a potential recipient. Given the increased risk of collision at *b* and unreliable nature of broadcast, the potential loss is assumed to be small.)

mine the duration of the current transmission from information in the control traffic and sleep until the end of the transmission. When the node wakes up, however, it does not have information about the state of the channel. For example, another neighbor may have begun a transmission, in which case the node should go back to sleep. In order to determine the duration of the current transmission, the node transmits a sequence of probe messages and awaits a response on the control channel. Similarly, if the node wishes to transmit, but another neighbor is now receiving, the node's RTS will evoke a busy tone response indicating the duration of the ongoing transmission. PAMAS is inherently conservative; a node sleeps only if it determines that it is possible to do so without affecting network capacity.

Simulation results suggest that PAMAS provides 10–70% reduction in the amount of energy a node spends receiving, assuming a 2:1 send-to-receive power consumption ratio. PAMAS is most effective in networks with high density and traffic load. It has no effect at all on the energy consumption of a quiescent network.

This technique is less applicable to network interfaces with high data transmission rates. If the time required for data transmission is short, then the time and energy required for the network interface to transition to the sleep state and back to the idle state outweigh the possible savings. Examples of low-power, low-data-rate transmitters are most often found in sensor network scenarios, where transmission rates of a few Kbps are not uncommon.

11.5 POWER CONTROL TECHNIQUES

The next class of techniques considered in this chapter allow nodes to modify their transmit power to increase network capacity and reduce energy consumption. Figure 11.4 shows an example of how the network topology is defined by the nodes' transmission radius. Low-power transmissions reduce contention and increase network capacity, while at the same time consuming less energy. This suggests that a route with a larger number of low-power hops may be more energy-efficient than one with fewer high-power hops. However, a route that contains more hops may experience a higher probability of network route failure or link-layer packet retransmission. These are further examples of the importance of cross-layer interaction in energy-efficient communication.

When evaluating the effectiveness of power-control techniques, it is important to consider whether the energy consumption model is realistic, particularly in its treatment of receiving. The extent to which the protocol operation depends the assumptions and accuracy of a particular radio propagation model is also extremely important and often difficult to evaluate in simulation.

The first two subsections below discuss the topology control problem briefly and the minimum energy routing problem in detail. The third subsection discusses problems that occur when multiple transmit powers are used in a network and describes some proposed solutions.

11.5.1 Topology Control

The topology-control problem is to assign per-node transmit powers that minimize the maximum transmit power used in the network, while still maintaining network connectivity. These methods are generally focused on increasing throughput by reducing interfer-

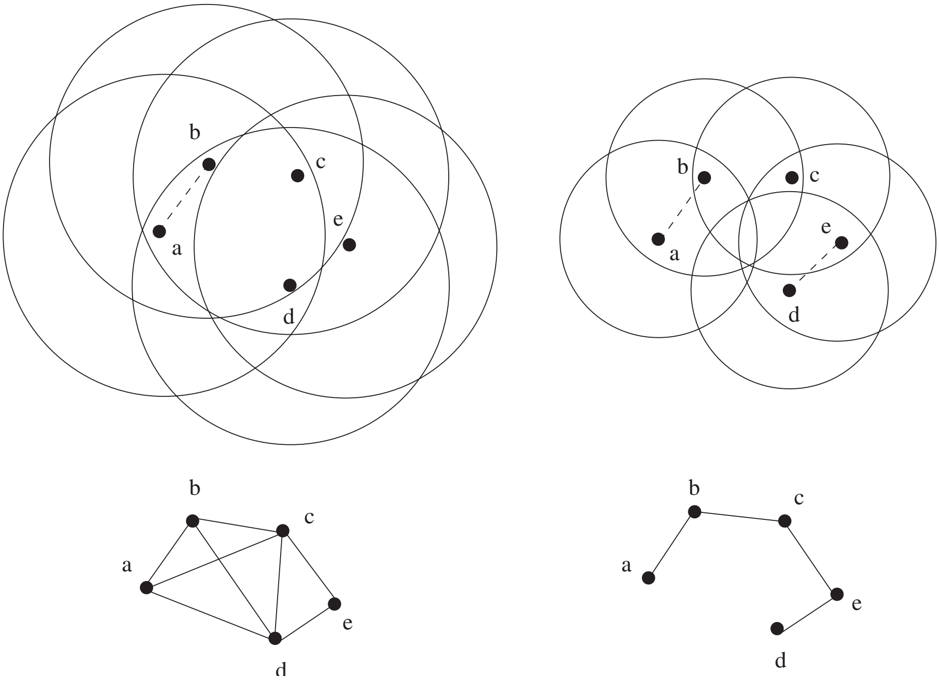


Figure 11.4. Transmit range determines the topology and capacity of this five-node network. (For simplicity, transmission and carrier sense range are assumed equal.) With $e_{i,j} \propto d_{i,j}^{-\alpha}$, a relay, e.g. $a \rightarrow b \rightarrow c$ may consume less energy than a direct route, for example, $a \rightarrow c$.

ence, with the associated reduction in energy consumption as a beneficial side effect. For that reason, this chapter will not emphasize the topic. Some simple topology control strategies are briefly discussed below.

In [29], link state topology information is used maintain a connected (or biconnected) topology. If a route update indicates that a link failure has occurred such that the network is no longer connected, the appropriate nodes increase their transmit power (using slotted backoff) until it is connected. This technique depends heavily on routing protocol performance, because changes in network connectivity can trigger further routing updates.

Alternatively, a variety of heuristics can be used. In [29], each node increases its transmit power until its node degree is sufficiently large, based on estimated node density, that the network is likely to be connected. In [39], each node modifies its transmit power until it has discovered at least one neighbor "in every direction," specifically in each cone of angle $\alpha \leq 2\pi/3$. Assuming an unobstructed and homogeneous environment, these neighbors' combined wireless coverage provides connectivity such that the node is guaranteed to be connected to the network.

Another variant of this problem is presented in [40]. Given a broadcast source node, the minimum-energy broadcast problem is to select a set of rebroadcasters and transmit powers such that the message is distributed to all nodes with minimum total energy cost. Unlike the topology-control problem, which has an optimal centralized solution, minimum-energy broadcast is believed to be NP-complete.

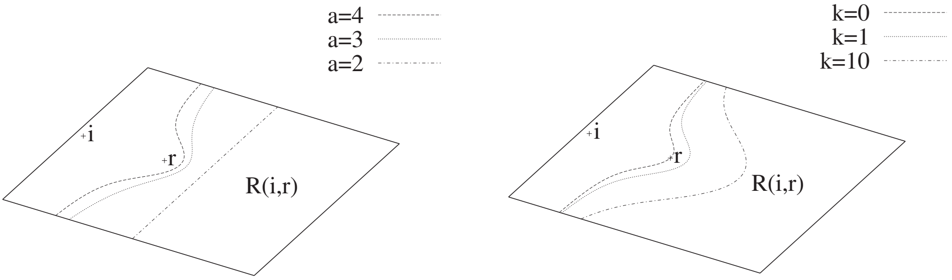


Figure 11.5. Relay region. For transmitter i and relay node r at unit distance ($d_{i,j}^{\alpha} = 1$), the relay region $R_i(r)$ is shown for several values of α ($k = 1$) (left), and of k ($\alpha = 4$) (right).

11.5.2 Minimum-Energy Routing

The minimum-energy routing problem is to minimize the total energy consumed in forwarding a packet from source to destination. Minimum-energy routing can exploit exponential path loss by forwarding traffic using a sequence of low power transmissions rather than a single direct transmission, as in Figure 11.5.

For successful transmission, the signal-to-noise plus interference ratio (SNIR) at the receiver must be greater than some threshold, which depends on the target bit error rate. In a basic path-loss model, received signal strength decreases exponentially with distance. The minimum transmit power required to transmit from node i to node j is

$$P_{\min_{ij}} \propto d_{i,j}^{\alpha} \qquad 2 \leq \alpha \leq 4$$

where exponent α depends on the environment.

The measurement data presented earlier show that it is necessary to account for energy consumed in both transmitting and receiving when evaluating the energy cost of a path. The former depends on the transmit power used at each hop, whereas the latter is roughly constant. If a relay node is added to a minimum-hop-count path, the energy saved though reduced transmit power must compensate for the energy consumed by the overhead of the extra transmit and receive operations. The cost of transmitting a packet over link (ij) is modeled as

$$e_{i,j} = d_{i,j}^{\alpha} + k$$

It is important to emphasize that in this context, the "distance" d, must be treated as notional for observed gain. The variable k reflects the fixed overhead.[5] Wireless propagation is very complex to model, due to effects of terrain and other obstacles. In general, it is not possible for a node to determine the transmit-power level required for nodes to communicate based on their positions. At a given transmit-power level, a node may be able to successfully transmit further in some directions (e.g., along an open corridor) than in others (e.g., through wall), as shown in Figure 11.6.

[5] Details of the MAC protocol can vary, for example, the receiver may transmit control traffic, but it is only important that all overhead factors are included. (For simplicity, a fixed packet length is assumed throughout.)

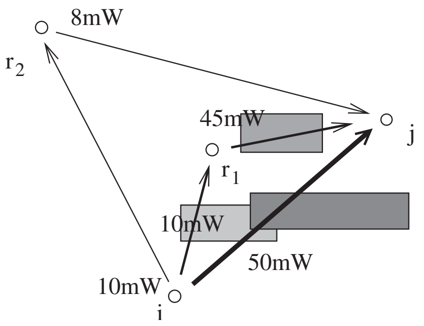


Figure 11.6. Here, $e_{i,j} \not\propto d^{\alpha}_{i,j}$ due to obstacles. Relay r_2 is the best candidate, despite its distant position.

The two examples of minimum-energy-routing techniques presented below are somewhat analogous to the proactive and reactive approaches to the ad hoc routing problem. The first is topology-driven and finds routes for all nodes. Because the minimum-energy route can change more frequently than a route requiring only connectivity, such solutions are appropriate for static or slowly changing networks. The second, which is more in the spirit of reactive routing protocols, assesses the energy efficiency of ongoing flows to improve their energy efficiency, adapting to the effects of node mobility.

11.5.2.1 Relay Regions. The minimum energy routing technique presented in [30, 19, 22] is based on the concept of a relay region, shown in Figure 11.5. For a given transmitter *i* and relay node *r*, the relay region $R_i(r)$ contains the set of receivers for which transmitting via relay node *r* reduces the total energy consumption. By combining the boundaries of the relay regions defined by taking each of its neighbors as a relay node, the transmitting node determines its "enclosure" region.

Any node that is in the set defined by the union of the relay regions is outside the enclosure: It can be more efficiently reached via some relay node. Nodes inside the enclosure are not in the relay region of any other node: Only these nodes are one-hop neighbors of the transmitting node in the minimum-energy topology. The neighbor sets defined by the enclosure of each transmitter define the minimum energy topology. Minimum-energy routes in this topology can be obtained using, for example, the distributed Bellman–Ford algorithm to find minimum-cost routes. They can also be approximated by running a routing protocol such as AODV over the minimum-energy topology.

Implementing this technique is apparently straightforward. Each node broadcasts neighbor discovery messages with increasing transmit-power levels, accumulating ACKs containing the positions of its neighbors. As new nodes are discovered, nodes that are not in the relay region of any other node are added to the neighbor set and nodes that are found to be in the relay region of another node are eliminated from it. This strategy is found to significantly reduce energy consumption in simulation environments providing uniform, deterministic transmit-power calculations. In practice, transmit power cannot be easily calculated from position information. To select minimum-energy routes, nodes must evaluate relay nodes based on the actual transmit-power levels required on the direct path and on both hops of the relay path. These issues emphasize the importance of incorporating realistic propagation behavior into simulation environments to study its impact on minimum-energy routing and to evaluate the effectiveness of various methods of estimating transmit-power levels.

11.5.2.2 *Adaptive Minimum-Energy Routing.* The power-aware routing optimization (PARO) protocol [11] represents a different approach to minimum-energy routing. Rather than exchanging connectivity information to determine a minimum-energy topology, nodes observe ongoing transmissions and nominate themselves as relay nodes for any pair of endpoints for which they provide more energy-efficient connectivity. This means that PARO uses observed gain rather than position information to evaluate routes. The protocol's adaptive operation also makes it natural to support nonstatic networks and reactive routing protocols.

In PARO, nodes eavesdrop on ongoing transmissions, using the advertised transmit-power level and observed signal strength to estimate the minimum transmit power required to communicate with each node that it overhears. If a node hears traffic from both endpoints of a link, it can compare the advertised transmit power being used to send the packet directly with its estimate of total transmit power required to send the packet using itself as a relay. If the energy savings is sufficiently high, the node sends a redirect message to each of the endpoints, updating their routing tables to use itself as a relay. To prevent several intermediate nodes from issuing redirect messages simultaneously, the message is sent using an adaptive backoff timeout that is inversely proportional to the optimization value of using the node as a redirector. This greedy selection of the redirector at each hop may not lead to an optimal route. In practice, however, most of the energy savings is gained in the first iteration of the protocol, especially if a realistic view is taken of the fixed receive costs.

Simulation studies of PARO reduces per-packet energy consumption by between one- and two-thirds compared to fixed power transmissions. The delivery ratio also remains good, except in the case where high mobility (i.e., short-lived links) is combined with low data rates (i.e., few opportunities to redirect). PARO also performed well compared to a simple link-state routing algorithm selecting minimum energy paths, due to the high routing protocol overhead. Experiments performed using nodes equipped with the Aironet 4800 IEEE 802.11 card were inconclusive. The authors suggest this may be due to the transmit-power granularity of the Aironet card, which only supports a few transmit-power levels.

A potential problem is that PARO requires that nodes eavesdrop to find opportunities for route optimization. Power-save protocols, by contrast, attempt to maximize the amount of time that nodes spend in the sleep state. In fully evaluating these protocols, it is important to consider the interaction between these aspects.

11.5.3 Problem of Multiple Transmit Powers

The presence of multiple transmit-power levels in a network poses a nontrivial problem, particularly in the CSMA case. Consider the example in Figure 11.7. If node a is transmitting a packet to nearby node b using low transmit power, the ongoing transmission may not be detected at node A. When node A begins transmitting a packet to more distant node B, it interferes with the transmission from a to b.

This problem can be mitigated by transmitting control traffic at the maximum power permitted in the network, as is suggested in [11] for the PARO protocol. Unfortunately, this reduces both the energy savings and the opportunity for increased network capacity from improved spatial reuse. Some alternative approaches are discussed below.

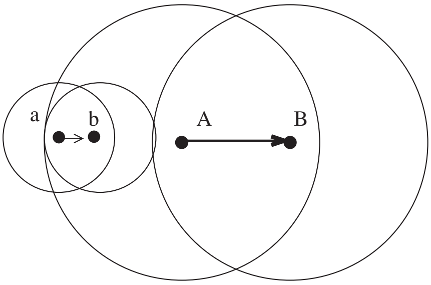


Figure 11.7. Problem of multiple transmit ranges (assuming carrier sense is equivalent to receive).

11.5.4 Adaptive Power Control

One approach to this problem is to use an adaptive power-control loop, similar to that found in CDMA systems. Adaptive power control for the IEEE 802.11 MAC is proposed in [1]. Each node maintains for each destination a weighted history of the received signal strength for successful transmissions and the threshold at which packet loss occurs. If the received signal strength indication is larger than this threshold, the transmission power is reduced (subject to the constraints of the control loop). If the MAC layer times out waiting for a response, it records this threshold signal strength and retransmits using a higher transmit power. (In the example above, nodes *a* and *b*, experiencing loss due to interference, would increase their transmit power until their control traffic properly suppressed interfering transmissions from node *A*.)

Simulation results show that adaptive power control is only moderately effective in a group-mobility environment. Energy consumed in transmitting and receiving traffic is reduced 10–20% and throughput is increased by about 15%, compared to an unmodified IEEE 802.11 MAC. Lesser savings are observed in a randomized environment. This is not unexpected: In group-mobility environments, power-control techniques can take advantage of the fact that most traffic is exchanged locally.

Because this technique cannot be used for broadcast transmissions, broadcast packets must be transmitted at full power. This means that the route discovery process (DSR [17] in these experiments), continues to discover routes with the smallest hop count, rather than minimum-energy routes. But because no additional hops are introduced by adaptive power control, the only source of overhead is the duplicate transmissions used for power-level adaptation.

11.5.5 Network Optimal Transmit Power

Another approach is to avoid the problems associated with per-node transmit powers altogether by determining an optimal common transmit-power level for the network; specifically the lowest transmit power level that maintains network connectivity. In [13] and [26], it is demonstrated analytically that a per-node throughput of $O(1/\sqrt{n \log n})$ is obtained using a common network-optimal transmit-power level, as compared with an upper bound of $O(1/\sqrt{n})$ with per-node optimal levels.

It is shown in [32] that the critical transmission power for a network is the transmission power associated with the highest-cost link in the minimum spanning tree of the network.

Simulation shows that, independent of the mobility model, the variability of this critical transmit power due to mobility is very high.

In practice, network interfaces usually provide a small number of fixed transmit-power levels, making precise determination of the critical transmit power unnecessary. The Common Power (COMPOW) Protocol [26], makes use of this fact. Each COMPOW node runs multiple instances of a proactive ad hoc routing protocol (DSDV [27]), one at each of the available transmit-power levels. The connectivity information is used to maintain the corresponding set of parallel routing tables. The lowest-power routing table that still includes all of the destinations that are included in the highest-power routing table is used as the default routing table and defines the common transmit power. An implementation of COMPOW is reported in [26], but no performance results have been published. In nonstatic networks, the overhead of multiple instances of the routing protocol may be large, as reported in [3].

11.6 MAXIMUM-LIFETIME ROUTING

The discussion of power-save protocols earlier in this chapter emphasized not only the importance of reducing idle-mode energy consumption, but also the importance of increasing the network lifetime by balancing energy consumption across the network. This section explores the problem of maximum-lifetime routing. The first part of this section discusses a number of route metrics used in selecting energy-aware routes. The second part briefly discusses some theoretical results before turning to issues involved in incorporating these metrics in practical routing protocols. The final part of this section provides an overview of some interesting alternative approaches, including one based on the electrochemical behavior of the battery.

11.6.1 Route-Selection Metrics

There are three metrics that are commonly included in energy-aware routing mechanisms. These are minimum-energy routing, max–min routing, and minimum-cost routing.

Minimum-energy routing, discussed in the previous section, minimizes the total energy consumed as a packet is forwarded on a route. (In [34], it is suggested that a minimum-energy route metric can also take into account the energy cost of link contention, but it is not clear how this could be done in practice.) Minimum-energy routing does not maximize network lifetime. Because the nodes' residual energy (remaining battery capacity) is not taken into account, nodes on the minimum-energy routes will suffer early failure due to their heavy forwarding load.

The max–min metric explicitly avoids this problem by selecting the route that maximizes the minimum residual energy of any node on the route. Routes selected using max–min metrics may be longer or have greater total energy consumption than the minimum-energy route. This increase in per-packet energy consumption tends to reduce the network lifetime, though max–min generally performs much better than minimum-energy routing.[6]

[6] A (somewhat pathological) topology in which max–min performs arbitrarily badly compared to minimum energy is shown in [23].

Minimum-cost routing minimizes the total *cost* of forwarding the packet at each node, selecting the route that minimizes the sum of the link costs c_{ij}. The shape of the cost function controls the extent to which the presence of a high-cost (i.e., low residual energy) node on a route deflects traffic from that route.

In general, the cost function is a monotonically increasing function, reflecting a node's increasing "reluctance" to forward traffic as its residual energy decreases. A simple example of a cost function for link (ij) is the reciprocal of the residual energy E at node i, $c_{ij} = 1/E_i$, or the normalized residual energy $c_{ij} = E_{\text{init}}/E_i$. In [34], the battery voltage z_i is used to define a cost function $1/z_i - z_{\text{crit}}$, which grows rapidly as the voltage nears the critical level. Alternatively, a residual energy-cost function can be based on a known discharge curve for a particular battery.

A capacity-cost function incorporates both the communication-energy cost e_{ij} for link (ij) and the residual energy at a node. One advantage of such a function is its ability to balance the objectives of maximizing energy reserves and minimizing forwarding cost. Examples include $c_{ij} = e_{ij}/E_i$ and its normalized counterpart $e_{ij}(E_{\text{init}}/E_i)$, which reflect the amount of traffic node i can forward if the route includes link (ij) with energy cost e_{ij}. Examples of capacity-cost functions that estimate the total cost of a route using only position and local cost information are found in [36].

These routing metrics have been widely studied in simulation [4, 5, 38]. With the exception of minimum energy and pure residual cost metrics, all of them appear to perform well. The results are roughly summarized in Table 11.2 and discussed in a little more detail below. It is not clear whether they would continue to hold true in more realistic simulation environments, which could produce networks with vulnerable routing bottlenecks and greater variability in route length and energy cost.

Simulations [4, 5] of a twenty node static network systematically compared total cost and max–min metrics for a variety of capacity and residual energy cost functions. In general, total normalized capacity was the most effective metric, especially if the residual energy is somewhat overweighted. Max–min metrics for both residual energy and capacity performed nearly as well. All these metrics obtained 84–96% of the optimum lifetime in the average case and 66–90% in the worst case. On average, minimum energy routing and total residual energy metrics performed poorly, achieving 30–50% of the optimum lifetime.

A capacity-based cost function is one way of jointly optimizing communication energy and residual energy. An alternative is to explicitly balance the minimum energy and max–min approaches, as in two protocols discussed below.

A conditional strategy, CMMBCR, is proposed in [38]. For each destination, if there is at least one route such that the residual energy of each node is greater than some thresh-

Table 11.2. Effectiveness of Various Route Metrics (Primarily from [5, 4]). Performance Differences Among the Better Metrics are Small

	Transmit cost e_{ij}	Residual energy E_i	Normal residual energy $\frac{E_i}{E_{\text{init}}}$	Capacity $\frac{e_{ij}}{E_i}$	Normal capacity $e_{ij}\ (E_{\text{init}}/E_i)$
Max–min	—	good	good	good	good
Total cost	very bad (minimum energy)	bad	bad	good	very good

old, the minimum energy route is chosen. If there is no such route, the route that maximizes the minimum residual energy is selected. Simulation results for a 30-node mobile network with fixed transmit power (i.e., the minimum-energy route is shortest path) shows how the choice of threshold value determines the expiration sequence. A max–min metric (i.e., a high threshold) gives the longest time to first node failure, whereas minimum-energy routing (i.e., a low threshold) gives the longest half-life. An intermediate value gives balanced behavior. This is precisely as expected: Because the max–min schemes use longer routes to avoid early node failure, the overall energy consumption increases and the average node lifetime is decreased. Overall, the differences among the metrics were only around 10-20%, as in the previous results. (Minimum-energy routing performed relatively better, perhaps because it is equivalent to shortest-path routing in this environment.)

An adaptive scheme, max-min$zP_{\min}$, is presented in [23]. In this case, a route that maximizes the minimum residual energy on the route is selected, as long as it consumes no more than $zP_{\min}$ energy, where $P_{\min}$ is the energy consumed by the minimum-energy route. Periodically, the estimated lifetime of each node is calculated based on its energy reserves and current rate of change. If the minimum node lifetime of the network has increased, z is increased, preferring max–min routes; otherwise z is decreased. Simulation results for static networks of up to 40 nodes indicate that the adaptive technique consistently achieves over 80% of the optimal lifetime.

11.6.2 Algorithms and Protocols

Maximum-lifetime routing is a difficult problem, even formulated in an abstract context. Given a set of known, constant rate flows and a fixed network topology, maximum-lifetime routing can be specified as a linear programming problem [4]; specifically, maximizing the minimum node lifetime, subject to multicommodity flow conservation. This lifetime is generally used as the optimum against which various approaches are compared. Given fixed per-node transmit power, maximum-lifetime routing is equivalent to maximum flow with node capacities. In [23], however, it is shown that for message sequences that are not known in advance, maximum-lifetime routing is an NP-hard problem. Moreover, there is no online algorithm having a constant competitive ratio with respect to an optimal offline algorithm.

The simulation results summarized in Table 11.2 generally indicate that there is relatively little difference in performance among the various metrics. But by themselves, metrics for selecting the optimal route do not provide maximum-lifetime routing. There must be some means by which candidate paths are identified, metrics computed, and flow directed to the appropriate path.

Because ad hoc routing protocols must support decentralized operation, maintaining up-to-date residual-energy information at all nodes for table-driven or distance-vector route calculations may not be a viable approach. In practice, it may also be difficult to isolate maximum-lifetime metrics from other aspects of ad hoc routing. Metrics such as route stability and QoS considerations for delay-sensitive flows are also important. Some approaches to these issues are discussed below.

In the simplest case, evaluation of route metrics can be incorporated into the route discovery phase of an on-demand routing protocol. Total or max–min values for the metric are accumulated as the route request traverses the network, and the optimal route is selected by the destination. However, route discovery techniques are usually optimized to minimize the bandwidth-intensive broadcast flooding operation and therefore only present a small subset of possible routes to the destination. Routes for long-lived flows may need to be periodically recomputed in order to respond to changes in residual capacity.

A flow-augmentation algorithm is presented in [4]. At each source, the shortest-cost path is recomputed for each unit of flow, λ, in order to respond to changes in battery capacity, with smaller λ giving best performance. Selecting the shortest-cost route for the chosen metric can, in principle, be done using some variant of the distributed Bellman–Ford algorithm. This may result in significant overhead in a dynamic, asynchronous network, however.

For networks with a known set of constant rate flows, flow redirection [4] is based on the observation that if network lifetime is maximized, then each path has the same lifetime. Otherwise, some flow could be redirected from the shortest lifetime path onto some other path. The algorithm uses feasible descent to determine the shortest lifetime path and redirect flow from that path to another one. Like max–min, this approach can exhibit arbitrarily poor performance, becoming trapped in a local minimum. In simulation, its performance lags somewhat behind the other approaches, especially in worst-case performance.

A hierarchical, zone-based variant of max–min$zP_{\min}$ is described in [23]. The nodes in each geographic zone estimate the capacity of the region, assuming max–min$ z P_{min}$ routing for traffic through the zone. This substantially reduces the overhead of distributing each node's energy status though the network. Moreover, the energy distribution within a zone changes more slowly than that of individual nodes. Traffic is routed using geographic forwarding, preferring high capacity zones rather than high-capacity nodes. The performance is comparable to that of max-min$zP_{\min}$, while supporting much larger networks.

In [36], route metrics are computed at each forwarding node based on local residual energy and link-cost information and extrapolated costs based on the distance from the forwarding node to the destination. This eliminates the need to distribute residual-energy information.

11.6.3 Alternative Approaches

Two alternative approaches to energy-aware routing integrate other important aspects of the system into the cost metrics used for route selection.

11.6.3.1 Battery-Efficient Routing. A routing scheme based on battery efficiency is presented in [7]. Batteries exhibit two electrochemical behaviors that can be exploited by a routing protocol. The first is the recovery effect, such that a bursty discharge pattern is more efficient than a constant-current discharge. The second is the rate-capacity effect, such that drawing even a small percentage of current impulses that exceed the rated current capacity of the battery significantly degrades battery performance. Battery-energy-efficient (BEE) routing uses a cost function that includes both a conventional max–min component and a penalty factor proportional to amount by which the transmit energy for a link exceeds the node mean. Simulation experiments that model battery discharge suggest that the time to network failure with BEE can be almost twice that obtained with minimum-energy routing.

11.6.3.2 Reliable Energy-Aware Routes. In addition to realistic energy consumption models, it is also important to use realistic models of channel error. The routing metrics presented in [2, 25] reflect the potential cost of retransmissions required to recover from link errors and achieve reliable end-to-end delivery. For example, minimum-energy routes can increase the probability of transmission error along the end-to-end path, due to the increased number of hops.

To find minimum-energy reliable routes, the cost function must reflect the expected cost of retransmissions at each link. Assuming the transmit power is fixed and bit-error failures on link (ij) are independent, the probability of retransmission is given as p_{ij}. If hop-by-hop retransmission is supported, the expected cost $c_{ij} = (e_{ij}/1 - p_{ij})$. If hop-by-hop retransmission is not supported, the cost c_{ij} is approximated by $e_{i,j}/(1 - p_{i,j})^L$, where L is the average path length in the network. In the case of fixed transmit power e_i, the received signal strength and the error probability p_{ij}, vary with link distance. Alternatively, a variable transmit power e_{ij} can be chosen such that the received signal strength, and thus the error probability p, are fixed.

The simulation study presented in [2] uses this retransmission-aware cost function to find minimum-energy routes. For hop-by-hop retransmissions and fixed transmit power, and for end-to-end retransmission and variable transmit power, retransmission-aware routing saves significant energy compared to minimum-energy or shortest-hop routing. In all cases, the TCP throughput also increased significantly.

From here, the development is analogous to that in the section above. In addition to computing minimum-energy routes, the link energy function can also be combined with residual energy E_i to define a capacity-cost function

$$c_{i,j} = \frac{e_{i,j}}{E_i(1 - p_{i,j})^L}$$

The simulation study presented in [25] uses this cost function to define max–min and conditional metrics MRPC and CMRPC, analogous to MMBCR and CMMBCR above. For a maximum error rate of 0.25, reliability-aware routing performed significantly better than its residual-energy counterpart, especially in dense networks. The conditional strategy shows little gain because, unlike MMBCR, the MRPC max–min metric already includes a minimum-energy routing element because the cost function includes the link energy.

11.7 CONCLUSION

This chapter has examined three current areas of research in energy-efficient communication in ad hoc networks. Power-save protocols attack the problem of high idle-state energy consumption by maximizing the amount of time nodes spend in the sleep state. Power control increases network capacity and reduces energy consumption by allowing nodes to determine the minimum transmit-power level required to maintain network connectivity and forward traffic with least energy cost. Maximum-lifetime routing selects paths that maximize network lifetime by balancing energy consumption across the nodes of the network.

In assessing the work that appears in this chapter, there are two themes that appear repeatedly. The first is the extent to which energy-efficient communication is a multifaceted problem. Attention to only one aspect of the problem, or to optimizing a single element of the protocol stack can lead to suboptimal performance with respect to the goal of maximizing node and battery lifetime. The limitations of minimum-energy routing and the importance of considering end-to-end reliability are examples of this. Moreover, it is not clear how the various approaches outlined here might interact if they were applied in the same network. For example, power-save protocols are most effective in a dense network, in which a small proportion of nodes remain awake to forward traffic. In minimum-energy topologies, on the other hand, it is precisely the network density that is reduced. Interactions with system- and application-level energy management techniques are also largely an open question [21].

The second theme is the central role that energy consumption and wireless propagation models play in the design and evaluation of energy-efficient systems. Direct measurements of wireless systems have been shown to provide energy consumption models that are useful in designing and evaluating energy management techniques. It is also shown that, due to the complex dependence of wireless propagation on terrain and other features, protocols must not rely on a predictive relationship between distance, transmit power, and connectivity.

Almost all of the results presented here are based on simulation, rather than direct experiment. This means that the results depend significantly on the wireless propagation and energy consumption models incorporated into the simulations. Often, the same models are used in both the design and evaluation of a protocol, which can be a source of confusion in interpreting results. This problem highlights the importance of developing simulation techniques (and even testbed environments [24]) that support complex and realistic analysis of techniques currently being developed for energy-efficient communication in wireless ad hoc networks.

REFERENCES

1. S. Agarwal, S. V. Krishnamurthy, R. H. Katz, and S. K. Dao. "Distributed power control in ad hoc wireless networks," in *Personal and Indoor Mobile Radio Communication (PIMRC),* 2001.
2. Suman Banerjee and Archan Misra. "Minimum energy paths for reliable communication in multi-hop wireless networks," in *Proceedings of Workshop on Mobile and Ad Hoc Networking and Computing (MobiHoc'02),* June 2002.
3. Josh Broch, David A. Maltz, David B. Johnson, Yih-Chun Hu, and Jorjeta Jetcheva. "A performance comparison of multi-hop wireless ad hoc network routing protocols," in *Proceedings of 4th Annual International Conference on Mobile Computing and Networking (MobiCom'98),* 1998.
4. Jae-Hwan Chang and Leandros Tassiulas. "Energy conserving routing in wireless ad-hoc networks," in *Proceedings of IEEE Infocom,* vol. 1, pp. 22–31, March 2000.
5. Jae-Hwan Chang and Leandros Tassiulas. "Maximum lifetime routing in wireless sensor networks," in *Proceedings of Advanced Telecommunications and Information Distribution Research Program (ATIRP'2000),* March 2000.
6. Benjie Chen, Kyle Jamieson, Hari Balakrishnan, and Robert Morris. Span: "An energy-ef_cient coordination algorithm for topology maintenance in ad hoc wireless networks. *ACM Wireless Networks Journal, 8*(5), 481–494, September 2002.
7. Carla-Fabiana Chiasserini and Ramesh R. Rao. "Routing protocols to maximize battery efficiency," in *Proceedings of IEEE Milcom,* October 2000.
8. Jean-Pierre Ebert, Brian Burns, and Adam Wolisz. "A trace-based approach for determining the energy consumption of a WLAN network interface," in *Proceedings of European Wireless,* pp. 230–236, February 2002.

9. Laura Marie Feeney. "An energy-consumption model for performance analysis of routing protocols for mobile ad hoc networks." *Journal of Mobile Networks and Applications (MONET), 6*(3), 239–250, June 2001.

10. Laura Marie Feeney and Martin Nilsson. "Investigating the energy consumption of a wireless network interface in an ad hoc networking environment," in *Proceedings of IEEE Infocom,* April 2001.

11. Javier Gomez, Andrew T. Campbell, Mahmoud Naghshineh, and Chatschik Bisdikian. "Conserving transmission power in wireless ad hoc networks," in *Proceedings of IEEE Conference on Network Protocols (ICNP'01),* November 2001.

12. Bluetooth Special Interest Group. Specification of the Bluetooth system. http://www.bluetooth.org.

13. P. Gupta and P. R. Kumar. "The capacity of wireless networks." *IEEE Transactions on Information Theory, 46*(2), 388–404, March 2000.

14. Erik Guttman. "Autocon_guration for IP networking: Enabling local communication." *IEEE Internet Computing, 5*(3), 81–86, May 2001.

15. Z. J. Haas and M. R. Pearlman. "Providing ad-hoc connectivity with the reconfigurable wireless networks," in *Ad Hoc Networking,* Charles Perkins (Ed.), Addison-Wesley, 2000.

16. IEEE Computer Society LAN MAN Standards Committee. *IEEE 802.11 Standard: Wireless LAN Medium Access Control and Physical Layer Specifications,* August 1999.

17. David B. Johnson, David A. Maltz, and Josh Broch. "DSR: The dynamic source routing protocol for multihop wireless ad hoc networks," in *Ad Hoc Networking,* Charles Perkins (Ed.), Addison-Wesley, 2000.

18. Oliver Kasten and Marc Langheinrich. "First experiences with bluetooth in the smart-its distributed sensor network," in *Proceedings of 10th International Conference on Parallel Architectures and Compilation Techniques (PACT'01) Workshop on Ubiquitous Computing and Communications (UCC01),* September 2001.

19. Ozge H. Koymen, Volkan Rodoplu, and Teresa H. Meng. "Throughput characteristics of a minimum energy wireless network," in *Proceedings of IEEE International Conference on Communications (ICC01),* June 2001.

20. Robin Kravets, Ken Calvert, and Karsten Schwan. "Power-aware communication for mobile computers," in *Proceedings of Sixth International Workshop on Mobile Multimedia Communications (MoMuc-6),* 1999.

21. Robin Kravets and P. Krishnan. "Power management techniques for mobile communication," in *Proceedings of 4th Annual International Conference on Mobile Computing and Networking (MobiCom'98),* 1998.

22. Li Li and Joseph Y. Halpern. "Minimum-energy mobile wireless networks revisited," in *Proceedings of IEEE International Conference on Communications (ICC01),* pp. 278–283, 2001.

23. Qun Li, Javed Aslam, and Daniela Rus. "Online power-aware routing in wireless ad-hoc networks," in *Proceedings of 7th Annual International Conference on Mobile Computing and Networking,* 2001.

24. H. Lundgren, D. Lundberg, J. Nielsen, E. Nordström, and C. Tschudin. "A large-scale testbed for reproducible ad hoc protocol evaluations," in *Proceedings of IEEE Wireless Communication and Networking Conference (WCNC'02),* March 2002.

25. Archan Misra and Suman Banerjee. MRPC: "Maximizing network lifetime for reliable routing in wireless environments," in *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC'02),* March 2002.

26. Swetha Narayanaswamy, Vikas Kawadia, R. S. Sreenivas, and P. R. Kumar. "Power control in ad-hoc networks: Theory, architecture, algorithm and implementation of the COMPOW protocol," in *Proceedings of European Wireless,* pp. 156–162, February 2002.

27. Charles Perkins and Pravin Bhagwat. "Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers," in *ACM Conference on Communications Architectures, Protocols and Applications (SIGCOMM'94),* pp. 234–244, 1994.

28. Charles E. Perkins and Elizabeth M. Royer. "Ad hoc on-demand distance vector routing," in *Ad Hoc Networking,* Charles Perkins (Ed.), Addison-Wesley, 2000.

29. Ram Ramanathan and Regina Hain. "Topology control of multihop wireless networks using transmit power adjustment," in *Proceedings of IEEE Infocom,* Vol. 2, pp. 404–413, 2000.

30. Volkan Rodoplu and Teresa H.-Y. Meng. "Minimum energy mobile wireless networks. *IEEE Journal on Selected Areas in Communications, 17*, 8, 1333– 1344, August 1999.

31. Elizabeth M. Royer and C.-K. Toh. "A review of current routing protocols for ad-hoc mobile wireless networks. *IEEE Personal Communications Magazine,* 46–55, April 1999.

32. Miguel Sanchez, Pietro Manzoni, and Zygmunt J. Haas. "Determination of critical transmission range in ad-hoc networks," in *Proceedings of Multiaccess Mobility and Teletraffic for Wireless Communications Workshop,* October 1999.

33. Suresh Singh and C.S. Raghavendra. "PAMAS—power aware multi-access protocol with signalling for ad hoc netowrks." *ACMComputer Communication Review,* July 1998.

34. Suresh Singh, MikeWoo, and C. S. Raghavendra. "Power-aware routing in mobile ad hoc networks," in *Proceedings of 4th Annual International Conference on Mobile Computing and Networking (MobiCom'98),* pp. 181–190, 1998.

35. M. Stemm and R. H. Katz. "Measuring and reducing energy consumption of network interfaces in hand-held devices. *IEICE Transactions on Communications, E80-B*, 8, 1125–1131, 1997.

36. Ivan Stojmenovic and Xu Lin. "Power aware localized routing in wireless networks." *IEEE Transactions on Parallel and Distributed Systems, 12*, 11, 1122–1133, November 2001.

37. The VINT Project. The ns manual. http://www.isi.edu/nsnam/ns.

38. C.-K. Toh. "Maximum battery life routing to support ubiquitous mobile computing in wireless ad hoc networks." *IEEE CommunicationsMagazine, 39*, 6, June 2001.

39. Roger Wattenhofer, Li Li, Paramvir Bahl, and Yi-Min Wang. "Distributed topology control for wireless multihop ad-hoc networks," in *Proceedings of IEEE Infocom,* pp. 1388–1397, April 2001.

40. JefferyWieselthier, Gam Nguyen, and Anthony Ephremides. "On the construction of energy-ef_cient broadcast and multicast trees in wireless networks," in *Proceedings of IEEE Infocom,* 2000.

41. Hagen Woesner, Jean-Pierre Ebert, Morten Schlager, and Adam Wolisz. "Power saving mechanisms in emerging standards for wireless LANs: The MAC level perspecitve." *IEEE Personal Communications, 5*, 3, 40–48, June 1998.

42. Jie Wu, Fei Dai, Ming Gao, and Ivan Stojmenovic. "On calculating power-aware connected dominating sets for ef_cient routing in ad hoc wireless networks." *IEEE/KICS Journal of Communications and Networks, 4*, 1, 59–70, March 2002.

43. Ya Xu. *Adaptive Energy Conservation Protocols for Wireless Ad Hoc Routing.* PhD thesis, University of Southern California, 2002.

44. Ya Xu, John Heidemann, and Deborah Estrin. "Adaptive energy-conserving routing for multihop ad hoc networks." Technical Report 527, USC/Information Sciences Institute, October 2000.

45. Ya Xu, John Heidemann, and Deborah Estrin. "Geography-informed energy conservation for ad hoc routing," in *Proceedings of 7th Annual International Conference on Mobile Computing and Networking,* pp. 70–84, July 2001.

CHAPTER 12

AD HOC NETWORKS SECURITY

PIETRO MICHIARDI and REFIK MOLVA

12.1 INTRODUCTION

An ad hoc network is a collection of wireless mobile hosts forming a temporary network without the aid of any established infrastructure or centralized administration. In such an environment, it may be necessary for one mobile host to enlist the aid of other hosts in forwarding a packet to its destination, due to the limited range of each mobile host's wireless transmissions. Mobile ad hoc networks (MANETs) do not rely on any fixed infrastructure but communicate in a self-organized way.

Security in a MANET is an essential component for basic network functions like packet forwarding and routing: network operation can be easily jeopardized if countermeasures are not embedded into basic network functions at the early stages of their design. Unlike networks using dedicated nodes to support basic functions like packet forwarding, routing, and network management, in ad hoc networks those functions are carried out by all available nodes. This very difference is at the core of the security problems that are specific to ad hoc networks. As opposed to dedicated nodes of a classical network, the nodes of an ad hoc network cannot be trusted for the correct execution of critical network functions.

If an *a priori trust relationship* exists between the nodes of an ad hoc network, entity authentication can be sufficient to assure the correct execution of critical network functions. A priori trust can only exist in a few special scenarios like military networks and corporate networks, where a common, trusted authority manages the network, and it requires tamper-proof hardware for the implementation of critical functions. Entity authentication in a large network, on the other hand, raises key management requirements. An environment where a common, trusted authority exists is called a *managed environment.*

Mobile Ad Hoc Networking. Edited by Basagni, Conti, Giordano, and Stojmenovic.
ISBN 0-471-37313-3 © 2004 Institute of Electrical and Electronics Engineers, Inc.

When tamper-proof hardware and strong authentication infrastructure are not available, for example, in an *open environment* where a common authority that regulates the network does not exist, any node of an ad hoc network can endanger the reliability of basic functions like routing. The correct operation of the network requires not only the correct execution of critical network functions by each participating node but it also requires that each node performs a fair share of the functions. The latter requirement seems to be a strong limitation for wireless mobile nodes in which power saving is a major concern. The threats considered in the MANET scenario are thus not limited to maliciousness; a new type of misbehavior called selfishness should also be taken into account to eliminate nodes that simply do not cooperate.

With *lack of a priori trust,* classical network security mechanisms based on authentication and access control cannot cope with selfishness, and cooperative security schemes seem to offer the only reasonable solution. In a cooperative security scheme, node misbehavior can be detected through the collaboration between a number of nodes, assuming that a majority of nodes do not misbehave.

The rest of the chapter is organized as follows. Section 12.2 presents the recent research that has been done in order to come up with secure routing protocols for ad hoc networks that cope with threats that are specific to the ad hoc environment. All of the presented secure protocols, however, do not take into account the node selfishness problem, which is detailed in Section 12.3. Recent solutions to combat the lack of node cooperation are presented in Section 12.3. The basic requirement of a large number of proposed security scheme is the presence of a key distribution mechanism managed by a trusted authority that takes part in the initialization phase of the network. Recent advances in order to provide an automated key management scheme that does not require the presence of any external infrastructure or bootstrap phase where keys are distributed are presented in Section 12.4. In Section 12.5, currently available security mechanisms implemented in the data link layer are detailed and analyzed. Furthermore, Section 12.5.3 focuses on a discussion about the relevance for the ad hoc environment of security mechanisms implemented in the data link layer.

12.2 SECURE ROUTING

Routing protocols for ad hoc networks are challenging to design. Wired network protocols (such as BGP) are not suitable for an environment where node mobility and network topology rapidly change. Such protocols also have high communication overhead because they send periodic routing messages even when the network is not changing. So far, researchers in ad hoc networking have studied the routing problem in a nonadversarial network setting, assuming a reasonably trusted environment. However, unlike networks using dedicated nodes to support basic functions like packet forwarding, routing, and network management, in ad hoc networks, those functions are carried out by all available nodes. This very difference is at the core of the increased sensitivity to node misbehavior in ad hoc networks, and the current proposed routing protocols are exposed to many different types of attacks.

Section 12.2.1 presents and classifies the threats that a misbehaving node can perpetrate to jeopardize the network operation. Recent research brought up the need to take into account node misbehavior at the early stages of the routing protocol design. Current efforts in secure routing protocol design are outlined and analyzed in Section 12.2.2.

12.2.1 Exploits Allowed by Existing Routing Protocols

Current ad hoc routing protocols are basically exposed to two different types of attacks: *active* attacks and *passive* attacks. An attack is considered to be active when the misbehaving node has to bear some energy costs in order to perform the threat, whereas passive attacks are mainly due to lack of cooperation, with the purpose of saving energy selfishly. Nodes that perform active attacks with the aim of damaging other nodes by causing network outages are considered to be *malicious* whereas nodes that perform passive attacks with the aim of saving battery life for their own communications are considered to be *selfish.*

Malicious nodes can disrupt the correct functioning of a routing protocol by *modifying* routing information, by *fabricating* false routing information, and by *impersonating* other nodes. Recent research studies [10] also brought up a new type of attack that goes under the name of *wormhole* attack. On the other side, selfish nodes can severely degrade network performance and eventually partition the network (X) by simply not participating to the network operation.

12.2.1.1 Threats Using Modification. Existing routing protocols assume that nodes do not alter the protocol fields of messages passed among nodes. Malicious nodes can easily cause traffic subversion and denial of service (DoS) by simply altering these fields. Such attacks compromise the *integrity* of routing computations. By modifying routing information, an attacker can cause network traffic to be dropped, be redirected to a different destination, or take a longer route to the destination, thus increasing communication delays.

12.2.1.2 Threats Using Impersonation. Since current ad hoc routing protocols do not *authenticate* routing packets, a malicious node can launch many attacks in a network by masquerading as another node (*spoofing*). Spoofing occurs when a malicious node misrepresents its identity in order to alter the vision of the network topology that a benign node can gather. As an example, a spoofing attack allows one to create loops in routing information collected by a node with the result of partitioning the network.

12.2.1.3 Threats Using Fabrication. The notation “fabrication” is used when referring to attacks performed by generating false routing messages. Such kinds of attacks can be difficult to identify as they come as valid routing constructs, especially in the case of fabricated routing error messages claiming that a neighbor can no longer be contacted.

12.2.1.4 Wormhole Attack. A more subtle type of active attack is the creation of a tunnel (or wormhole) in the network between two colluding malicious nodes linked through a private network connection. This exploit allows a node to short-circuit the normal flow of routing messages, creating a virtual vertex cut in the network that is controlled by the two colluding attackers.

12.2.1.5 Lack of Cooperation. A selfish node that wants to save battery life for its own communication can endanger the correct network operation by simply not participating in the routing protocol or by not executing the packet forwarding (this attack is also known as the black hole attack) . Current ad hoc routing protocols cannot cope with the selfishness problem and network performances severely degrade as a result.

12.2.2 Secure Routing Protocols

Current efforts toward the design of secure routing protocols are mainly oriented to reactive (on-demand) routing protocols such as DSR [12] or AODV [13], in which a node attempts to discover a route to some destination only when it has a packet to send to that destination. On-demand routing protocols have been demonstrated to perform better with significantly lower overheads than proactive routing protocols in many scenarios since they are able to react quickly to topology changes, yet are able to reduce routing overhead in periods or areas of the network in which changes are less frequent. It is possible to find, however, interesting security solutions for proactive routing protocols that are worthwhile mentioning.

Common to the secure routing protocols proposed in the literature is the type of attack they address: major efforts are made to find countermeasures against *active attacks* performed by malicious nodes that aim at intentionally disrupt the routing protocol execution, whereas the selfishness problem is not addressed. Furthermore, the prerequisite for all the available solutions is a *managed* environment. In such scenarios, nodes wishing to communicate may be able to exchange initialization parameters beforehand; for example, within the security of a dedicated network where session keys may be distributed or through a trusted third party.

In the following, the major secure routing protocols for ad hoc networks will be outlined and analyzed.

12.2.2.1 SRP. The Secure Routing Protocol (SRP) [1], proposed by Papadimitratos and Haas, is conceived of as an extension that can be applied to a multitude of existing *reactive* routing protocols. SRP combats attacks that disrupt the route discovery process and guarantees the acquisition of correct topological information: a node initiating a route discovery is able to identify and discard replies providing false routing information or avoid receiving them.

The underlying assumption is the existence of a *security association* (SA) between the source node (S) and the destination node (T). The trust relationship could be instantiated, for example, by knowledge of the public key of the other communicating end. The two nodes can negotiate a shared secret key ($K_{S,T}$) and then, using the SA, verify that the principal that participated in the exchange was indeed the trusted node.

SRP copes with noncolluding *malicious* nodes that are able to modify (corrupt), replay, and fabricate routing packets. Based on the dynamic source routing protocol (DSR) SRP requires the addition of a six-word header containing unique identifiers that tag the discovery process, and a message authentication code (MAC). In order to initiate a route request (RREQ), the source node has to generate a MAC using a keyed hash algorithm that accepts as input the entire IP header, the basis protocol RREQ packet, and the shared key $K_{S,T}$.

The intermediate nodes that relay the RREQ toward the destination measure the frequencies of queries received from their neighbors in order to regulate the query propagation process: each node maintains a priority ranking that is inversely proportional to the queries rate. A node that maliciously pollutes network traffic with unsolicited RREQs will be served last (if not ignored) because of its low priority ranking.

Upon reception of a RREQ, the destination node verifies the *integrity* and *authenticity* of the RREQ by calculating the keyed hash of the request fields and comparing them with the MAC contained in the SRP header. If the RREQ is valid, the destination initiates a route replay (RREP) using the SRP header, the same way the source did when initiating the request. The source node discards replays that do not match with pending query identifiers and checks the integrity using the MAC generated by the destination.

The basic version of SRP is subject to route cache poisoning attacks: routing information gathered by nodes that operate in promiscuous mode in order to improve the efficiency of the DSR protocol could be invalid because they were fabricated by malicious nodes. The authors propose two alternative designs of SRP that uses an Intermediate Node Reply Token (INRT). INRT allows intermediate nodes that belong to the same group that share a common key (K_G) to validate RREQ and provide valid RREP messages.

SRP suffers also from the lack of a validation of route maintenance messages: route errors packets are not verified. However, in order to minimize the effects of fabricated error messages, SRP source-routes error packets along the prefix of the route reported as broken; as a consequence, the source node can verify that the provided route error feedback refers to the actual route and is not generated by a node that is not even part of the route. A malicious node can harm only the route it belongs to.

Assuming that the neighbor discovery mechanism maintains information on the binding of the medium-access control and the IP addresses of nodes, SRP has proven to be essentially immune to IP spoofing [1].

SRP is, however, not immune to the wormhole attack: two colluding malicious nodes can misroute the routing packets on a private network connection and alter the network topology that a benign node can collect.

12.2.2.2 ARIADNE.

Hu, Perrig, and Johnson developed Ariadne, an *on-demand* secure ad hoc routing protocol based on DSR that withstands node compromise and relies only on highly efficient *symmetric* cryptography. Ariadne guarantees that the target node of a route discovery process can authenticate the initiator, that the initiator can authenticate each intermediate node on the path to the destination present in the RREP message, and that no intermediate node can remove a previous node in the node list in the RREQ or RREP messages.

As for the SRP protocol, Ariadne needs some mechanism to bootstrap authentic keys required by the protocol. In particular, each node needs a shared secret key ($K_{S,D}$ is the shared key between a source S and a destination D) with each node it communicates with at a higher layer, an authentic TESLA [3, 4] key for each node in the network, and an authentic "Route Discovery Chain" element for each node for which this node will forward RREQ messages.

Ariadne provides point-to-point *authentication* of a routing message using a message authentication code (MAC) and a shared key between the two parties. However, for authentication of a broadcast packet such as RREQ, Ariadne uses the TESLA broadcast authentication protocol. Ariadne copes with attacks performed by *malicious* nodes that modify and fabricate routing information, with attacks using impersonation, and, in an advanced version, with the wormhole attack. Selfish nodes are not taken into account.

In Ariadne, the basic RREQ mechanism is enriched with eight fields used to provide authentication and integrity to the routing protocol:

<ROUTE REQUEST, initiator, target, id, time interval, hash chain, node list, MAC list>

The initiator and target are set to the address of the initiator and target nodes, respectively. As in DSR, the initiator sets the ID to an identifier that it has not recently used in initiating a route discovery. The time interval is the TESLA time interval at the pessimistic expected arrival time of the request at the target, accounting for clock skew. The initiator of the request then initializes the hash chain to $MAC_{KS,D}$ (initiator, target, ID, time interval) and the node list and MAC list to empty lists.

When any node *A* receives a RREQ for which it is not the target, the node checks its local table of <initiator, id> values from recent requests it has received, to determine if it has already seen a request from this same route discovery. If it has, the node discards the packet, as in DSR. The node also checks whether the time interval in the request is valid: that time interval must not be too far in the future, and the key corresponding to it must not have been disclosed yet. If the time interval is not valid, the node discards the packet. Otherwise, the node modifies the request by appending its own address (*A*) to the node list in the request, replacing the hash chain field with H [A, *hash chain*], and appending a MAC of the entire REQUEST to the MAC list. The node uses the TESLA key K_{Ai} to compute the MAC, where *i* is the index for the time interval specified in the request. Finally, the node rebroadcasts the modified RREQ, as in DSR.

When the target node receives the RREQ, it checks the validity of the request by determining that the keys from the time interval specified have not been disclosed yet, and that the hash chain field is equal to

$$H\ [\eta_n\ ,\ H\ [\eta_{n-1}\ ,\ H\ [\ .\ .\ .\ ,\ H\ [\eta_1\ ,\ MAC_{KSD}\ (\text{initiator, target, id, time interval})\ ]\ .\ .\ .\ ]\ ]\ ]$$

where η_i is the node address at position i of the node list in the request, and where n is the number of nodes in the node list. If the target node determines that the request is valid, it returns a RREP to the initiator, containing eight fields:

<ROUTE REPLY, target, initiator, time interval, node list, MAC list, target MAC, key list>

The target, initiator, time interval, node list, and MAC list fields are set to the corresponding values from the RREQ, the target MAC is set to a MAC computed on the preceding fields in the reply with the key KDS , and the key list is initialized to the empty list. The RREP is then returned to the initiator of the request along the source route obtained by reversing the sequence of hops in the node list of the request.

A node forwarding a RREP waits until it is able to disclose its key from the time interval specified, then it appends its key from that time interval to the key list field in the reply and forwards the packet according to the source route indicated in the packet. Waiting delays the return of the RREP but does not consume extra computational power.

When the initiator receives a RREP, it verifies that each key in the key list is valid, that the target MAC is valid, and that each MAC in the MAC list is valid. If all of these tests succeed, the node accepts the RREP; otherwise, it discards it.

In order to prevent the injection of invalid route errors into the network fabricated by any node other than the one on the sending end of the link specified in the error message, each node that encounters a broken link adds TESLA authentication information to the route error message, such that all nodes on the return path can authenticate the error. However, TESLA authentication is delayed, so all the nodes on the return path buffer the error but do not consider it until it is authenticated. Later, the node that encountered the broken link discloses the key and sends it over the return path, which enables nodes on that path to authenticate the buffered error messages.

Ariadne is also protected from a flood of RREQ packets that could lead to a cache poisoning attack. Benign nodes can filter out forged or excessive RREQ packets using *route* *discovery chains,* a mechanism for authenticating route discovery, allowing each node to rate-limit discoveries initiated by any other node. The authors present two different approaches that can be found in [2].

ARIADNE is immune to the wormhole attack only in its advanced version: using the TIK (TESLA with Instant Key disclosure) protocol that allows for very precise time synchronization between the nodes of the network, it is possible to detect anomalies in routing traffic flows in the network.

12.2.2.3 ARAN. The ARAN secure routing protocol proposed by Dahill, Levine, Royer, and Shields [5] is conceived of as an on-demand routing protocol that detects and protects against malicious actions carried out by third parties and peers in the ad hoc environment. ARAN introduces *authentication,* message *integrity,* and *nonrepudiation* as part of a minimal security policy for the ad hoc environment and consists of a preliminary certification process, a mandatory end-to-end authentication stage, and an optional second stage that provides secure shortest paths.

ARAN requires the use of a trusted certificate server (T): before entering the ad hoc network, each node has to request a certificate signed by T. The certificate contains the IP address of the node, its public key, a timestamp of when the certificate was created, and a time at which the certificate expires, along with the signature by T. All nodes are supposed to maintain fresh certificates with the trusted server and must know T's public key.

The goal of the first stage of the ARAN protocol is for the source to verify that the intended destination was reached. In this stage, the source trusts the destination to choose the return path. A source node, *A*, initiates the route discovery process to reach the destination *X* by broadcasting to its neighbors a route discovery packet called RDP:

$$[\mathrm{RDP};\ \mathrm{IP_X};\ cert_A;\ \mathrm{N_A};\ \mathrm{t}]\mathrm{K_{A-}}$$

The RDP includes a packet type identifier ("RDP"), the IP address of the destination ($\mathrm{IP_X}$), *A*'s certificate ($cert_A$), a nonce $\mathrm{N_A}$, and the current time t, all signed with A's private key. Each time A performs route discovery, it monotonically increases the nonce.

Each node records the neighbor from which it received the message. It then forwards the message to each of its neighbors, signing the contents of the message. This signature prevents spoofing attacks that may alter the route or form loops. Let *A*'s neighbor be *B*. It will broadcast the following message:

$$[[\mathrm{RDP};\ \mathrm{IP_X};\ cert_A;\ \mathrm{N_A};\ \mathrm{t}]\mathrm{K_{A-}}\ ]\mathrm{K_{B-}};\ cert_B$$

Nodes do not forward messages for which they have already seen the ($\mathrm{N_A}$; $\mathrm{IP_A}$) tuple. The IP address of A is contained in the certificate, and the monotonically increasing nonce facilitates easy storage of recently received nonces.

Upon receiving the broadcast, *B*'s neighbor *C* validates the signature with the given certificate. *C* then rebroadcasts the RDP to its neighbors, first removing B's signature:

$$[[\mathrm{RDP};\ \mathrm{IP_X};\ cert_A;\ \mathrm{N_A};\ \mathrm{t}]\mathrm{K_{A-}}\ ]\mathrm{K_{C-}};\ cert_C$$

Eventually, the message is received by the destination, *X*, which replies to the first RDP that it receives for a source and a given nonce. There is no guarantee that the first RDP received traveled along the shortest path from the source. The destination unicasts a

Reply (REP) packet back along the reverse path to the source. Let the first node that receives the RDP sent by *X* be node *D*. *X* will send to *D* the following message:

$$[\mathrm{REP};\ \mathrm{IP_A}\ ;\ cert_X\ ;\ \mathrm{N_A}\ ;\ \mathrm{t}]\mathrm{K_{X-}}$$

The REP includes a packet-type identifier ("REP"), the IP address of A, the certificate belonging to *X*, and the nonce and associated timestamp sent by *A*. Nodes that receive the REP forward the packet back to the predecessor from which they received the original RDP. All REPs are signed by the sender. Let *D*'s next hop to the source be node *C*. *D* will send to *C* the following message:

$$[[\mathrm{REP};\ \mathrm{IP_A}\ ;\ cert_X\ ;\ \mathrm{N_A}\ ;\ \mathrm{t}]\mathrm{K_{X-}}\ ]\mathrm{K_{D-}}\ ;\ cert_D$$

C validates *D*'s signature, removes the signature, and then signs the contents of the message before unicasting the following RDP message to *B*:

$$[[\mathrm{REP};\ \mathrm{IP_A}\ ;\ cert_X\ ;\ \mathrm{N_A}\ ;\ \mathrm{t}]\mathrm{K_{X-}}\ ]\mathrm{K_{C-}}\ ;\ cert_C$$

A node checks the signature of the previous hop as the REP is returned to the source. This avoids attacks in which malicious nodes instantiate routes by impersonation and replay of *X*'s message. When the source receives the REP, it verifies that the correct nonce was returned by the destination as well as the destination's signature. Only the destination can answer an RDP packet. Other nodes that already have paths to the destination cannot reply for the destination. Although other protocols allow this networking optimization, ARAN removes several possible exploits and cuts down on the reply traffic received by the source by disabling this option.

The second stage of the ARAN protocol guarantees in a secure way that the path received by a source initiating a route discovery process is the shortest. Similarly to the first stage of the protocol, the source broadcasts a *Shortest Path Confirmation* (SPC) message to its neighbors. The SPC message is different from the RDP message only in two additional fields that provide the destination *X* certificate and the encryption of the entire message with *X*'s public key (which is a costly operation). The onion-like signing of messages combined with the encryption of the data prevents nodes in the middle from changing the path length because doing so would break the integrity of the SPC of the packet.

Also, the route maintenance phase of the ARAN protocol is secured by digitally signing the route error packets. However, it is extremely difficult to detect when error messages are *fabricated* for links that are truly active and not broken. Nevertheless, because messages are signed, malicious nodes cannot generate error messages for other nodes. The nonrepudiation provided by the signed error message allows a node to be verified as the source of each error message that it sends.

As with any secure system based on cryptographic certificates, the key revocation issue has to be addressed in order to make sure that expired or revoked certificates do not allow the holder to access the network. In ARAN, when a certificate needs to be revoked, the trusted certificate server T sends a broadcast message to the ad hoc group that announces the revocation. Any node receiving this message rebroadcasts it to its neighbors. Revocation notices need to be stored until the revoked certificate would have expired normally. Any neighbor of the node with the revoked certificate needs to reform routing as necessary to avoid transmission through the now untrusted node. This method is not fail-safe. In some cases, the untrusted node that is having its certificate revoked may be the sole connection between two parts of the ad hoc network. In this case, the untrusted node may not forward the notice of revocation for its certificate, resulting in a partition of the network, as nodes that have received the revocation notice will no longer forward messages through the untrusted node, whereas all other nodes depend on it to reach the rest of the network. This only lasts as long as the untrusted node's certificate would have otherwise been valid, or until the untrusted node is no longer the sole connection between the two partitions. At the time that the revoked certificate should have expired, the untrusted node is unable to renew the certificate, and routing across that node ceases. Additionally, to detect this situation and to hasten the propagation of revocation notices, when a node meets a new neighbor, it can exchange a summary of its revocation notices with that neighbor; if these summaries do not match, the actual signed notices can be forwarded and rebroadcast to restart propagation of the notice.

The ARAN protocol protects against exploits using *modification, fabrication,* and *impersonation,* but the use of asymmetric cryptography makes it a very costly protocol to use in terms of CPU and energy usage. Furthermore, ARAN is not immune to the *wormhole* attack

12.2.2.4 SEAD. Hu, Perrig, and Johnson developed a *proactive* secure routing protocol called SEAD [7], based on the Destination-Sequenced Distance Vector protocol (DSDV). In a proactive (or periodic) routing protocol, nodes periodically exchange routing information with other nodes in attempt to have each node always know a current route to all destinations. SEAD was inspired by the DSDV-SQ version of the DSDV protocol. The DSDV-SQ version of the DSDV protocol has been shown to outperform other DSDV versions in previous ad hoc networks simulations [8, 9].

SEAD deals with attackers that *modify* routing information broadcast during the update phase of the DSDV-SQ protocol; in particular, routing can be disrupted if the attacker modifies the sequence number and the metric field of a routing table update message. *Replay attacks* are also taken into account.

In order to secure the DSDV-SQ routing protocol, SEAD makes use of efficient *one-way hash chains* rather than relying on expensive asymmetric cryptography operations. However, like the other secure protocols presented in this chapter, SEAD assumes some mechanism for a node to distribute an authentic element of the hash chain that can be used to authenticate all the other elements of the chain. As a traditional approach, the authors suggest to ensure the key distribution by relying on a trusted entity that signs public key certificates for each node; each node can then use its public key to sign a hash chain element and distribute it.

The basic idea of SEAD is to authenticate the sequence number and metric of a routing table update message using hash chains elements. In addition, the receiver of SEAD routing information also authenticates the sender, ensuring that the routing information originates form the correct node.

To create a one-way hash chain, a node chooses a random initial value $x \in \{0,1\}^{\rho}$, where ρ is the length in bits of the output of the hash function, and computes the list of values $h_0, h_1, h_2, h_3, \ldots, h_n$, where $h_0 = x$, and $h_i = \mathrm{H}(h_{i-1})$ for $0 < i \leq n$, for some n. As an example, given an authenticated h_i value, a node can authenticate h_{i-3} by computing $\mathrm{H}(\mathrm{H}(\mathrm{H}(h_{i-3})))$ and verifying that the resulting value equals *hi*.

Each node uses a specific authentic (i.e., signed) element from its hash chain in each routing update that it sends about itself (metric 0). Based on this initial element, the one-way hash chain provides authentication for the lower bound on the metric in other routing updates for that node. The use of a hash value corresponding to the sequence number and metric in a routing update entry prevents any node from advertising a route to some destination claiming a higher sequence number than that destination's own current sequence number. Likewise, a node cannot advertise a route better than those for which it has received an advertisement, since the metric in an existing route cannot be decreased due to the one-way nature of the hash chain.

When a node receives a routing update, it checks the authenticity of the information for each entry in the update using the destination address, the sequence number, and the metric of the received entry, together with the latest prior *authentic* hash value received from that destination's hash chain. Hashing the received elements the correct number of times (according to the prior authentic hash value) assures the authenticity of the received information if the calculated hash value and the authentic hash value match.

The source of each routing update message in SEAD must also be authenticated since, otherwise, an attacker may be able to create routing loops through the *impersonation* attack. The authors propose two different approaches to provide node authentication: the first is based on a broadcast authentication mechanism such as TESLA, and the second is based on the use of Message Authentication Codes, assuming a shared secret key between each couple of nodes in the network.

SEAD does not cope with *wormhole* attacks, although the authors propose, as in the ARIADNE protocol, to use the TIK protocol to detect the threat.

12.2.3 Notes on the Wormhole Attack

The wormhole attack is a severe threat against ad hoc routing protocols that is particularly challenging to detect and prevent. In a wormhole attack, a malicious node can record packets (or bits) at one location in the network and tunnel them to another location through a private network shared with a colluding malicious node. Without some mechanism to defend them against the wormhole attack, most existing ad hoc routing protocols would be unable to find consistent routes to any destination, severely disrupting communication.

A dangerous threat can be perpetrated if a wormhole attacker tunnels all packets through the wormhole honestly and reliably since no harm seems to be done; the attacker actually seems to provide a useful service in connecting the network more efficiently. However, when an attacker forwards only routing control messages and not data packets, communication may be severely damaged. As an example, when used against an on-demand routing protocol such as DSR, a powerful application of the wormhole attack can be mounted by tunneling each RREQ message directly to the destination target node of the request. This attack prevents routes more than two hops long from being discovered because RREP messages would arrive to the source faster than any other replies or, worse, RREQ messages arriving from nodes next to the destination other than the attacker would be discarded as already seen.

Hu, Perrig, and Johnson propose an approach to detect a wormhole based on *packet leashes* [10]. The key intuition is that by authenticating either an extremely precise timestamp or location information combined with a loose timestamp, a receiver can determine if the packet has traversed a distance that is unrealistic for specific network technology used.

Temporal leashes rely on extremely precise time synchronization and extremely precise timestamps in each packet. The travel time of a packet can be approximated as the difference between the receive time and the timestamp. Given the precise time synchronization required by temporal leashes, the authors propose efficient broadcast authenticators based on symmetric primitives. In particular, they extend the TESLA broadcast authentication protocol to allow the disclosure of the authentication key within the packet that is authenticated.

Geographical leashes are based on location information and loosely synchronized clocks. If the clocks of the sender and the receiver are synchronized within a certain threshold and the velocity of any node is bounded, the receiver can compute an upper bound on the distance between the sender and itself and use it to detect anomalies in the traffic flow. Under certain circumstances, however, bounding the distance between the sender and the receiver cannot prevent wormhole attacks: when obstacles prevent communication between two nodes that would otherwise be in transmission range, a distance-based scheme would still allow wormholes between the sender and the receiver. To overcome this problem, in a variation of the geographical leashes, the receiver verifies that every possible location of the sender can reach every possible location of the receiver based on a radio propagation model implemented in every node.

In some special cases, wormholes can also be detected through techniques that do not require precise time synchronization or location information. As an example, it would be sufficient to modify the routing protocol used to discover the path to a destination so that it could handle multiple routes; a verification mechanism would then detect anomalies when comparing the metric (e.g., number of hops) associated with each route. Any node advertising a path to a destination with a metric considerably lower than all the others could raise the suspect of a wormhole.

Furthermore, if the wormhole attack is performed only on routing information while dropping data packets, other mechanisms can be used to detect this misbehavior. When a node does not correctly participate in the network operation by not executing a particular function (e.g., packet forwarding) a collaborative monitoring technique can detect and gradually isolate misbehaving nodes. Lack of cooperation and security mechanisms used to enforce node cooperation in network operation is the subject of the next section.

12.3 COOPERATION ENFORCEMENT IN MOBILE AD HOC NETWORKS

As opposed to networks using dedicated nodes to support basic networking functions like packet forwarding and routing, in ad hoc networks these functions are carried out by all available nodes in the network. There is no reason, however, to assume that the nodes in the network will eventually cooperate with one another since network operation consumes energy, a particularly scarce resource in a battery powered environment like MANET. The new type of node misbehavior that is specific to ad hoc networks is caused by lack of cooperation and goes under the name of *node selfishness.* A selfish node does not directly intend to damage other nodes with active attacks (mainly because performing active attacks can be very expensive in terms of energy consumption) but it simply does not cooperate in network operation, saving battery life for its own communications.

Damage caused by selfish behavior cannot be underestimated: a simulation study present in the literature [11] shows the impact of a selfish behavior in terms of global network throughput and global communication delay when the DSR routing protocol is used. The simulation results show that even a small percentage of selfish nodes present in the network leads to a severe degradation of performance. Furthermore, any security mechanism that tries to enforce cooperation among the nodes should focus not only on one particular function, but on both the routing and the packet forwarding function. As an example, if a source routing mechanism such as DSR is used, any node that does not participate to the routing protocol cannot claim to participate in the packet forwarding function since it cannot appear in any route, meaning that it will never be asked to relay packets for other nodes of the network.

The node selfishness problem has only recently been addressed by the research community, and there are still few mechanisms provided to combat such misbehavior. Mechanisms that enforce node cooperation in a MANET can be divided into two categories: the first is currency-based (see Section 12.3.1) and the second uses a local monitoring technique (see Sections 12.3.2 and 12.3.3). Currency-based systems are simple to implement but rely on a tamperproof hardware. The main drawback of this approach is in the difficulty of establishing how the virtual currency has to be exchanged, making their use not realistic in a practical system. Cooperative security schemes based on local monitoring seem to offer the most suitable solution to the selfishness problem. Every node of the MANET monitors its local neighbors, evaluating for each of them a metric that is directly related to the nodes' behavior. Based on that metric, a selfish node can be gradually isolated from the network. The main drawback of this approach is related to the absence of a mechanism that securely identifies the nodes of the network: any selfish node could elude the cooperation enforcement mechanism and get rid of its bad reputation just by changing its identity.

In the following, the main research efforts toward the solution of the node selfishness problem are presented.

12.3.1 Nuglets

In [14],] Buttyan and Hubaux present two important issues targeted specifically at the ad hoc networking environment: first, endusers must be given some incentive to cooperate in the network operation (especially to relay packets belonging to other nodes); second, endusers must be discouraged from overloading the network. The solution presented in their paper consists of the introduction of a virtual currency (which they call Nuglets) used in every transaction. Two different models are described: the Packet Purse Model and the Packet Trade Model. In the Packet Purse Model, each packet is loaded with nuglets by the source and each forwarding host takes out nuglets for its forwarding service. The advantage of this approach is that it discourages users from flooding the network, but the drawback is that the source needs to know exactly how many nuglets it has to include in the packet it sends. In the Packet Trade Model, each packet is traded for nuglets by the intermediate nodes: each intermediate node buys the packet from the previous node on the path. Thus, the destination has to pay for the packet. The direct advantage of this approach is that the source does not need to know how many nuglets need to be loaded into the packet. On the other hand, since the packet generation is not charged for, malicious flooding of the network cannot be prevented. There are some further issues that have to be solved: concerning the Packet Purse Model, the intermediate nodes are able to take out more nuglets than they are supposed to; concerning the Packet Trade Model, the intermediate nodes are able to deny the forwarding service after taking out nuglets from a packet.

12.3.2 CONFIDANT

The acronym given to the cooperation mechanism proposed by Buchegger and Le Boudec stands for “Cooperation Of Nodes, Fairness In Dynamic Ad-hoc NeTworks” [15, 16] and it detects malicious nodes by means of observation or reports about several types of attacks, thus allowing nodes to route around misbehaving nodes and to isolate them. CONFIDANT works as an extension to a routing protocol such as Dynamic Source Routing (DSR).

Nodes have a monitor for observations, reputation records for first-hand and trusted second-hand observations about routing and forwarding behavior of other nodes, trust records to control trust given to received warnings, and a path manager to adapt their behavior according to reputation and to take action against malicious nodes. The term reputation is used to evaluate routing and forwarding behavior according to the network protocol, whereas the term trust is used to evaluate participation in the CONFIDANT metaprotocol.

The dynamic behavior of CONFIDANT is as follows. Nodes monitor their neighbors and change the reputation accordingly. If they have reason to believe that a node misbehaves, they can take action in terms of their own routing and forwarding and they can decide to inform other nodes by sending an ALARM message. When a node receives such an ALARM either directly or by promiscuously listening to the network, it evaluates how trustworthy the ALARM is based on the source of the ALARM and the accumulated ALARM messages about the node in question. It can then decide whether to take action against the misbehaved node in the form of excluding routes containing the misbehaved node, reranking paths in the path cache, reciprocating by noncooperation, or forwarding an ALARM about the node.

The first version of CONFIDANT was, despite the filtering of ALARM messages in the trust manager, vulnerable to concerted efforts of spreading wrong accusations. This problem has been addressed by the use of Bayesian statistics for classification and the exclusion of liars.

Simulations with nodes that do not participate in the forwarding function have shown that CONFIDANT can cope well, even if half of the network population acts maliciously. Further simulations concerning the effect of second-hand information and slander have shown that slander can effectively be prevented while still retaining a significant detection speed-up over using merely first-hand information.

The limitations of CONFIDANT lie in the assumptions for detection-based reputation systems. Events have to be observable and classifiable for detection, and reputation can only be meaningful if the identity of each node is persistent, otherwise it is vulnerable to spoofing attacks.

12.3.3 CORE

The security scheme proposed by Michiardi and Molva [18, 19], stimulates node cooperation by a collaborative monitoring technique and a reputation mechanism. Each node of the network monitors the behavior of its neighbors with respect to a requested function and collects observations about the execution of that function. As an example, when a node initiates a Route Request (e.g., using the DSR routing protocol) it monitors that its neighbors process the request, whether with a Route Reply or by relaying the Route Request. If the observed result and the expected result coincide, then the observation will take a positive value, otherwise it will take a negative value.

Based on the collected observations, each node computes a reputation value for every neighbor using a sophisticated reputation mechanism that differentiates between subjective reputation (observations), indirect reputation (positive reports by others), and functional reputation (task-specific behavior), which are weighted for a combined reputation value. The formula used to evaluate the reputation value avoids false detections (caused, for example, by link breaks) by using an aging factor that gives more relevance to past observations: frequent variations on a node behavior are filtered. Furthermore, if the function that is being monitored provides an acknowledgement message (e.g., the Route Reply message of the DSR protocol), reputation information can also be gathered about nodes that are not within the radio range of the monitoring node. In this case, only positive ratings are assigned to the nodes that participated to the execution of the function in its totality.

The CORE mechanism resists attacks performed using the security mechanism itself: no negative ratings are spread between the nodes, so that it is impossible for a node to maliciously decrease another node's reputation. The reputation mechanism allows the nodes of the MANET to gradually isolate selfish nodes: when the reputation assigned to a neighboring node decreases below a predefined threshold, service provision to the misbehaving node will be interrupted. Misbehaving nodes can, however, be reintegrated into the network if they increase their reputation by cooperating in the network operation.

As for the other security mechanism based on reputation, the CORE mechanism suffers from spoofing attacks: misbehaving nodes are not prevented from changing their network identity, allowing the attacker to elude the reputation system. Furthermore, no simulation results prove the robustness of the protocol even if the authors propose an original approach based on game theory in order to come up with a formal assessment of the security properties of CORE.

12.3.4 Token-Based Cooperation Enforcement

In the approach presented by Yang, Meng, and Lu [20], each node of the ad hoc network has a token in order to participate in network operations, and its local neighbors collaboratively monitor it to detect any misbehavior in routing or packet forwarding services. Upon expiration of the token, each node renews its token via its multiple neighbors; the period of validity of a node's token is dependent on how long it has stayed and behaved well in the network. A well-behaving node accumulates its credit and renews its token less and less frequently as time evolves.

The security solution proposed by the authors is composed of four closely interacted components: *neighbor verification,* which describes how to verify whether each node in the network is a legitimate or malicious node; *neighbor monitoring,* which describes how to monitor the behavior of each node in the network and detect occasional attacks from malicious nodes; *intrusion reaction,* which describes how to alert the network and isolate the attackers; and the *security enhanced routing protocol,* which explicitly incorporates the security information collected by the other components into the ad hoc routing protocol.

Concerning the token issuing/renewal phase, the authors assume a global secret/public key pair SK/PK, where PK is well known by every node of the network. SK is shared by k neighbors who collaboratively sign the token requested or renewed by local nodes. Token verification follows three steps: 1) identity match between the node's ID and the token ID, 2) validity time verification, and 3) issuer signature. If the token verification phase fails, the corresponding node is rejected from the network and both routing and data packets are dropped for that node.

Routing security relies on the redundancy of routing information rather than cryptographic techniques. The routing protocol that the authors use as a basis is the Ad hoc On demand Distance Vector protocol (AODV), which is extended in order to detect false routing update messages by comparing routing information gathered from different neighboring nodes. Packet forwarding misbehavior is also detected using a modified version of the watchdog technique presented in [17], bypassing the absence of any source route information by adding a next-hop field in the routing messages.

The proposed solution presents some drawbacks: the bootstrap phase needed to generate a valid collection of partial tokens that will be used by a node to create its final token has some limitations. For example, the number of neighbors necessary to complete the signature of every partial token has to be at least k, suggesting the use of such security mechanism in rather large and dense ad hoc networks. On the other hand, the validity period of a token increases proportionally to the time during which the node behave well. This interesting feature has less impact if node mobility is high. Frequent changes in the local subset of the network that shares a key for issuing valid tokens can cause high computational overhead, not to mention the high traffic generated by issuing/renewing a token, suggesting that the token-based mechanism is more suitable in ad hoc networks where node mobility is low. Spoofing attacks, whereby a node can request more than one token by claiming different identities, are not taken into account even if the authors suggest that MAC addresses can be sufficient for node authentication purposes.

12.4 KEY MANAGEMENT

Providing security support for ad hoc networks is challenging for a number of reasons: wireless networks are susceptible to security attacks ranging from passive eavesdropping to active interfering and DoS attacks; occasional break-ins in a large-scale mobile network are inevitable over a large time interval; ad hoc networks provide no infrastructure support; mobile nodes may constantly leave or join the network; mobility-induced wireless links breakage/reconnection and wireless channel errors make timely communications over multiple hops highly unreliable; and a scalable solution is a must for a large-scale network. However, the provision of basic security services such as authentication, confidentiality, integrity, and nonrepudiation is critical in order to deploy the mobile wireless ad hoc technology in commercial and military environments.

Authentication services specific to the ad hoc environment have been recently studied by the research community in order to come up with a fully self-organized architecture to overcome the limitations intrinsic to the secure routing protocols that have been presented in Section 12.2.2.

The basic assumption adopted by some secure routing protocols such as SRP is the existence of an a priori security association between all the communicating nodes of the network. The limitations introduced by this approach range from the need for a managed environment, such as a common authority that precharges all the mobile terminals with a secret key shared by every couple of communicating nodes, to scalability problems.

Other secure routing protocols (such as Ariadne) rely on an initialization phase during which a well-known trusted third party (TTP) issues public key certificates used to authenticate (together with the private key of each certificate holder) hash chain elements that will be subsequently used to provide some low-cost (in terms of CPU usage) authentication services. In this case, the use of such a secure protocol is not limited to the managed environment, and the open environment can be targeted. Indeed, it is not necessary for the mobile nodes that form the ad hoc network to be managed by the same authority that provides the initial authentication setup. However, the bootstrap phase requires an external infrastructure, which also has to be available during the lifetime of the ad hoc network to provide revocation services for certificates that have expired or been explicitly revoked.

Current efforts to provide scalable, fully self-organized public key infrastructure and authentication services can be classified into two categories: one based on a PGP-like architecture and one based on the polynomial secret sharing technique. These are presented in the next sections.

12.4.1 Self-Organized Public-Key Management Based on PGP

Capkun, Buttyan, and Hubaux propose a fully self-organized public key management system that can be used to support security of ad hoc network routing protocols [21]. The suggested approach is similar to PGP [22] in the sense that users issue certificates for each other based on their personal acquaintances. However, in the proposed system, certificates are stored and distributed by the users themselves, unlike in PGP, where this task is performed by on-line servers (called certificate directories). In the proposed self-organizing public-key management system, each user maintains a *local certificate repository.* When two users want to verify the public keys of each other, they merge their local certificate repositories and try to find appropriate certificate chains within the merged repository that make the verification possible.

The success of this approach very much depends on the construction of the local certificate repositories and on the characteristics of the certificate graphs. By a certificate graph is meant to be a graph whose vertices represent public keys of the users and the edges represent public key certificates issued by the users. The authors investigate several repository construction algorithms and study their performance. The proposed algorithms take into account the characteristics of the certificate graphs in a sense that the choice of the certificates that are stored by each mobile node depends on the connectivity of the node and its certificate graph neighbors.

More precisely, each node stores in its local repository several directed and mutually disjoint paths of certificates. Each path begins at the node itself, and the certificates are added to the path such that a new certificate is chosen among the certificates connected to the last node on the path (initially the node that stores the certificates). The new certificate leads to the node that has the highest number of certificates connected to it (i.e., the highest vertex degree). The authors call this algorithm the *Maximum Degree Algorithm,* as the local repository construction criterion is the degree of the vertices in a certificate graph.

In a second, more sophisticated, algorithm that is called the *Shortcut Hunter Algorithm,* certificates are stored in the local repositories based on the number of shortcut certificates connected to the users. The shortcut certificate is a certificate that, when removed from the graph, makes the shortest path between two users previously connected by this certificate strictly larger than two.

When verifying a certificate chain, the node must trust the issuer of the certificates in the chain for correctly checking that the public key in the certificate indeed belongs to the node identification (ID) named in the certificate. When certificates are issued by the mobile nodes of an ad hoc network instead of trusted authorities, this assumption becomes unrealistic. In addition, there may be malicious nodes that issue false certificates. In order to alleviate these problems, the authors propose the use of authentication metrics [23]: it is not enough to verify a node ID key binding via a single chain of certificates. The authentication metric is a function that accepts two keys (the verifier and the verified node) and a certificate graph and returns a numeric value corresponding to the degree of authenticity of the key that has to be verified: one example of an authentication metric is the number of disjoint chains of certificates between two nodes in a certificate graph.

The authors emphasize that before being able to perform key authentication, each node must first build its local certificate repository, which is a relatively expensive operation (in terms of bandwidth and time). However this initialization phase must be performed rarely and once the certificate repositories have been built, any node can perform key authentication using only local information and the information provided by the targeted node. It should also be noted that local repositories become obsolete if a large number of certificate are revoked, as then the certificate chains are no longer valid. The same comment applies in the case when the certificate graph changes significantly. Furthermore, PGP-like schemes are more suitable for small communities because that the authenticity of a key can be assured with a higher degree of trustworthyness. The authors propose the use of authentication metrics to alleviate this problem: this approach however provides only probabilistic guarantees and is dependent on the characteristics of the certificate graph on which it operates. The authors also carried out a simulation study showing that for the certificate graphs that are likely to emerge in self-organized systems, the proposed approach yields good performance both in terms of the size of the local repository stored in each node and scalability.

12.4.2 Ubiquitous and Robust Authentication Services Based on Polynomial Secret Sharing

In [24], Luo and Lu present a mechanism that provides ubiquitous authentication service availability by taking a *certificate-based* approach. In the proposed scheme, any two communicating nodes can establish a temporary trust relationship via globally verifiable certificates. With a *scalable threshold sharing* of the certificate signing key, certification services (issuing, renewal, and revocation) are distributed among each node in the network: a single node holds just a share of the complete certificate signing key. Although no single node has the power of providing full certification services, multiple nodes in a network locality can collaboratively provide such services.

The authors propose a *localized trust model* to characterize the localized nature of security concerns in large ad hoc wireless networks. When applying such trust model, an entity is trusted if any k trusted entities claim so. These k trusted entities are typically the neighboring nodes of the entity. A locally trusted entity is globally accepted and a locally distrusted entity is regarded untrustworthy anywhere. k is a system-wide parameter that sets the global acceptance criteria and should be honored by each entity in the system.

The basic assumptions that are necessary for the security mechanism to function properly are: 1) each node has a unique nonzero identifier (ID), such as its MAC layer address; 2) each node has some one-hop discovery mechanism; 3) each node has at least k one-hop legitimate neighboring nodes, or the network has a minimum density of well-behaving nodes; 4) each node is equipped with some detection mechanism to identify misbehaving nodes among its one-hop neighborhood; 5) the mobility is characterized by a maximum node-moving speed $S_{\max}$.

In the security architecture proposed [24], each node carries a certificate signed by the shared certificate-signing key SK, and the corresponding public key PK is assumed to be well known by all the nodes of the network, so that certificates are globally verifiable. Nodes without valid certificates will be isolated; that is, their packets will not be forwarded by the network. Essentially, any node without a valid certificate is treated the same as adversaries. When a mobile node moves to a new location, it exchanges certificates with its new neighbors and goes through a mutual authentication process to build trust relationships. Neighboring nodes with such trust relationships help each other to forward and route packets. They also monitor each other to detect possible attacks and break-ins. Specific monitoring algorithms and mechanisms are left to each individual node's choice. When a node requests a signed certificate by the coalition of k nodes, each of the certificate issuing nodes checks its record on the requesting node. If the record shows that the requestor is a well-behaving legitimate node, it returns a partial certificate by applying its share of SK. Otherwise, the request is dropped. By collecting k partial certificates, the requesting node combines them together to generate the full new certificate as if it were issued from a certification authority server. A misbehaving or broken node that is detected by its neighbors will be unable to get a new certificate.

The security of the certificate-signing key SK is protected by a *k*-threshold polynomial sharing mechanism. However, this technique requires a bootstrapping phase in which a "dealer" has to privately send to each node its share of the SK. The authors propose a scalable initialization mechanism that they called "self-initialization." In this case, the dealer is only responsible for initializing the very first *k* nodes, no matter how large the network is. The initialized nodes collaboratively initialize other nodes; repeating this procedure, the network progressively self-initializes itself. The same mechanism is applied when new nodes join the network.

Certificate revocation is also handled by the proposed architecture and an original approach to handle roaming adversaries is presented. Without this additional mechanism, any misbehaving node that moves to a location of the network where its new neighbors have no information in their monitoring records about the attacker could get a new valid certificate. Roaming nodes are defeated with the flooding of "accusation" messages that travel in the network and inform distant nodes about the behavior of a suspect node. Accusation messages are accepted only if they come from well-behaving nodes and have a specific time-to-live (TTL) field that is calculated based on the maximum node speed specified in the assumptions.

The main drawbacks of the proposed architecture range from the necessity for an external, trusted dealer that initializes the very first *k* nodes of a coalition, to the choice of the system-wide parameter *k*. To cope with the first problem, the authors propose to use a distributed RSA key-pair generation [25] for the very first *k* nodes. On the other hand, no practical solutions are presented to cope with the strong assumption that every node of the network has at least *k* trusted and noncompromised neighbors. This limitation makes the proposed architecture useless for all the nodes that are located at the perimeter of the ad hoc network. More over, the authors assume that any new node that joins the system already has an initial certificate. Initial certificates can be obtained in two ways: every node may be issued an initial certificate by an offline authority, or every new node may use any coalition of *k* neighbors to issue the initial certificate via a collaborative admission control mechanism. These problems reduce the effectiveness of the proposed architecture as a fully self-organized infrastructure.

12.5 SECURITY MECHANISMS IN LAYER 2

This section presents data-link-layer security solutions that are suitable for MANET. The most prevalent solutions are the security mechanisms that function as part of 802.11 and Bluetooth specifications. Further to a detailed overview of their specifications, the weaknesses of each solution is analyzed. The relevance of these solutions with respect to the security requirements of MANET are then discussed.

12.5.1 Wired Equivalent Privacy (WEP)

The first security scheme provided in the series of IEEE 802.11 standards is Wired Equivalent Privacy (WEP), specified as part of the 802.11b Wireless Fidelity (Wi-Fi) standard [26]. WEP was originally designed to provide security for wireless local area networks (WLAN) with a level of protection that is similar to the one expected in wired LANs. The latter enjoy security and privacy due to their physical security mechanisms like building access control. Physical security mechanisms, unfortunately, do not prevent eavesdropping and unauthorized access in case of wireless communications. WEP thus aims at covering the lack of physical security of WLANs with security mechanisms based on cryptography. Unfortunately, WEP suffers from various design flaws and some exposure in the underlying cryptographic techniques that seriously undermine its security claims.

12.5.1.1 WEP Security Mechanisms. WEP security mechanisms include data encryption and integrity. Both mechanisms are handled simultaneously for each frame, as illustrated in Figure 12.1.

To prepare a protected frame, first an integrity check value (ICV) of the frame payload is computed using a cyclic redundancy check (CRC) function. The cleartext payload concatenated with ICV is then encrypted using a bit-wise exclusive-or operation with a keystream as long as the payload concatenated with ICV. The keystream is a pseudorandom bit stream generated by the RC4 [28] algorithm from a 40-bit secret key prepended

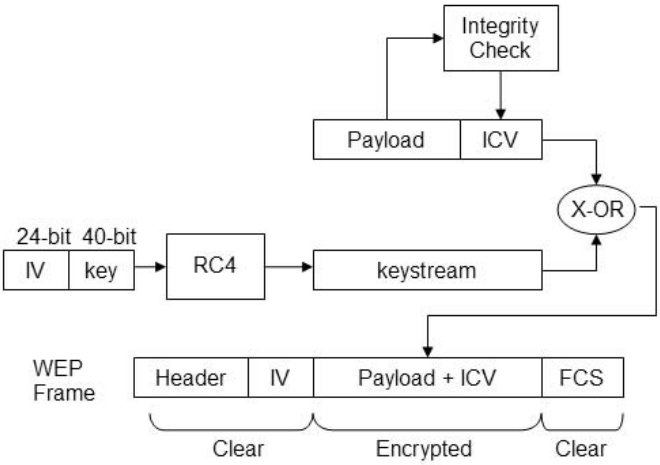


Figure 12.1. WEP frame security mechanisms.

with a 24-bit initialization value (IV). The resulting protected frame includes the cleartext frame header, the cleartext IV, the result of the encryption, and a cleartext frame-check sequence field.

The recipient of a WEP frame first generates the keystream with RC4 using the shared secret key and the IV value retrieved from the received frame. The resulting keystream is exclusive-ored with the encrypted field of the frame to decrypt the payload and the ICV. The integrity of the payload is then checked by comparing the integrity check computed on the cleartext payload with the ICV resulting from the decryption.

The secret key can either be a default key shared by all the devices of a WLAN or a pair-wise secret shared only by two communicating devices. Since WEP does not provide any support for the exchange of pair-wise secret keys, the secret key must be manually installed on each device.

12.5.1.1.1 Security Problems in WEP. WEP suffers from many design flaws and some weaknesses in the way the RC4 cipher is used Data encryption in WEP is based on an approximation of the "one-time pad" [28] algorithm that can guarantee perfect secrecy under some circumstances. Like WEP encryption, one-time pad encryption consists of the bit-wise exclusive-or between a binary plaintext message and a binary keystream as long as the message. The secrecy of the resulting ciphertext is perfect provided that each new message is encrypted with a different secret random keystream. The secrecy is not guaranteed when the keystream is reused or its values can be predicted. Hence, a first class of attacks on WEP exploit possible weaknesses in WEP's keystream generation process that make the secret keystream easily predictable or cause its reuse.

The first type of exposure is due to the likeliness of keystream reuse between a pair of communicating devices. Using the same secret key, the only variation in the input to the keystream generator is due to the variation in the IV. Since the IV is a 24-bit value sent in cleartext, the reuse of a keystream can be easily detected. The reuse of a keystream is also very likely because of the small set of possible IV values that can be exhausted in a few hours in busy traffic between two nodes. This type of exposure gets even worse if some care is not taken during the implementation of the standard: some products set the IV to a constant value (0 or 1) at the initialization of the encryption process for each frame sequence. The second type of exposure is due to the use of a 40-bit secret that is highly vulnerable to exhaustive search with current computational power.

WEP data encryption is also exposed through an advanced attack that takes into account the characteristics of the RC4 algorithm [29] and drastically reduces the set of possible keystream values based on the attacker's ability to recover the first byte of encrypted WEP payload.

Another class of exposure on WEP concerns the data integrity mechanism using CRC in combination with the one-time pad encryption. Encryption using exclusive-or operation is transparent with respect to modification, in that flipping bits of the encrypted message causes flipped bits on the same positions of the cleartext values resulting from decryption. As opposed to a cryptographically secure hash function, an integrity check computed with CRC yields predictable changes on the ICV with respect to single-bit modifications on the input message. Combining the transparency of exclusive-or with the predictable modification property of CRC, an attacker can flip bits on well-known positions of an encrypted WEP payload and on the corresponding positions of the encrypted ICV so that the resulting cleartext payload is modified without the modification being detected by the recipient. It should be noted that the transparent modification of the WEP payload does not require the knowledge of the secret payload value since the attacker only needs to know the location of some selected fields in the payload to force tampering with their value.

The last weakness of WEP is the lack of key management that is a potential exposure to most attacks exploiting manually distributed secrets shared by large populations.

12.5.1.1.2 A New Proposal. To address the shortcomings of WEP, IEEE has set up a special Task Group I (TGi) in charge of designing the new security architecture as part of the forthcoming version of the standard called 802.11i . To cope with brute force attacks, TGi has already proposed to include a 128-bit RC4 seed of which 104 bits are secret. TGi also proposed a long-term architecture based on the IEEE 802.1x standard, which itself is based on the IETF's Extensible Authentication Protocol (EAP). IEEE 802.1x has a flexible design supporting various authentication modes. However, the new proposal based on 802.1x already suffers from problems like lack of data integrity for wireless frames and lack of mutual authentication.

12.5.2 Bluetooth Security Mechanisms

The Bluetooth specification [27] includes a set of security profiles defined for the application layer in the so-called service-level security and security profiles for the data-link layer. Both types of profiles rely on key management, authentication, and confidentiality services based on cryptographic security mechanisms implemented in the data-link layer. Each Bluetooth device stands for an independent party from the point of view of the security protocols. In each device, security mechanisms use a set of basic components:

- The device address (BD_ADDR): a-48 bit address defined by the IEEE that is unique for each Bluetooth device
- A 128-bit authentication key
- A 128-bit symmetric data encryption key
- A random number (RAND) generated by a pseudorandom or (physical) random number generator

12.5.2.1 Key Management. Bluetooth key management services provide each device with a set of symmetric cryptographic keys required for the initialization of a secret channel with another device, the execution of an authentication protocol, and the exchange of encrypted data with another device.

12.5.2.1.1 Key Hierarchy. The key hierarchy of Bluetooth includes two generic key types:

1. The *link key* that is shared by two or more parties and used as a key-encrypting key (KEK) to encrypt other keys during key exchange, or as a seed to generate other keys
2. The *encryption key* that is a shared data-encryption key (DEK)

Both the link key and the encryption key are 128-bit symmetric keys. The link key can further be qualified as an *initialization key,* a *unit key,* a *combination key,* or a *master key.*

When two devices need to communicate using link-level security and have no prior engagement, they establish a secure channel based on the *initialization key.* This channel is then used by the communicating devices to establish a *semipermanent* link key that will be used several times to assure further key exchange between the devices. The *initialization key* is not used beyond the first key exchange. Each communicating device generates the *initialization key* using a pseudorandom number generator seeded with a secret personal identification number (PIN) entered by the user of each device and the value of RAND exchanged between the devices. In order for two communicating devices to generate the same value for the initialization key, the same PIN value must therefore be entered on both devices.

Based on the key generation and storage capabilities of each communicating device, the *semipermanent* shared link key can either be a *unit key* or a *combination key* depending on the key generation and storage capabilities of communicating devices. The *unit key* is a device specific key that is generated during the initialization of each device. Its value changes rarely. It is generated using a pseudorandom number generator seeded with RAND and BD_ADDR (device address). The *combination key* is a pairwise key computed by two communicating devices based on a device-specific key generated by each device. In order to compute a *combination key,* first each device generates a device specific key based on RAND and BD_ADDR; the resulting keys are then exchanged between the pair of devices using the secure channel encrypted with the *initialization key.* The *combination key* is then derived by each of the pair of devices based on a simple combination of the two device-specific keys.

The type of the link key to be used on a pairwise connection between two communicating devices is negotiated during link establishment. If one of the devices has restricted storage, then this device's unit key is used as the pairwise link key, with obvious drawbacks due to the widespread disclosure of a semipermanent device-specific key. If both communicating devices have sufficient computing (key generation) and storage capabilities, then they choose to use a combination key as a pairwise link key that has a different value for each pair of entities.

In a master–slave scenario, a short-lived link key called the *master key* can be used between the master device and the slave devices. The lifetime of a *master key* is limited to the duration of the master–slave session. The *master key* is generated by the master device using a pseudorandom number generator seeded by RAND and PIN. The resulting key is distributed through a channel secured under the *initialization key* to each slave the master wants to share the *master key* with.

The *encryption key* is generated by a pair of devices that share a link key using a pseudorandom number generator seeded by the link key, the random number RAND generated by one of the devices and transmitted to the other device prior to encrypted data exchange, and the secret Authenticated Ciphering Offset (ACO) generated by each device during the authentication process.

12.5.2.2 Authentication. The Bluetooth authentication scheme is based on a challenge–response protocol as depicted in Figure 12.2. When device A wants to authenticate device B using this protocol, A generates a random number (RAND) and sends it to B as a challenge, then both devices compute a result (SRES) using the authentication algorithm E1 with RAND, the link key, and the device address of B. B then sends A a SRES as the response to A's challenge RAND. B is successfully authenticated if the resulting SRES computed by A matches its SRES. During authentication, each device also obtains the Au-

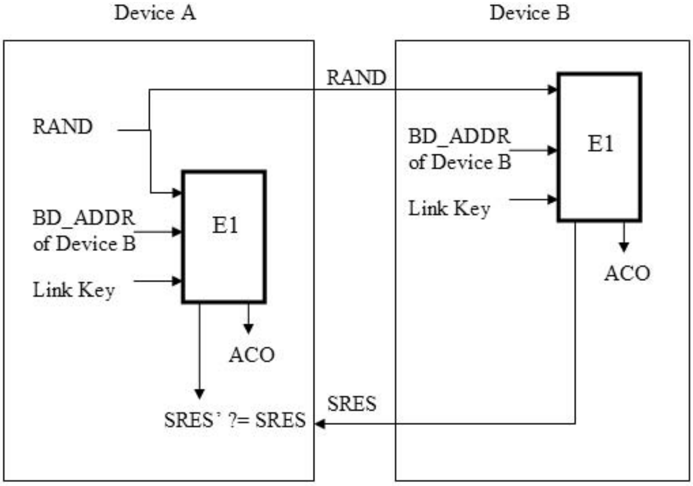


Figure 12.2. Authentication of Device B by Device A.

thenticated Ciphering Offset (ACO) generated by the authentication algorithm E1. The ACO value is further used to generate the data-encryption key that will be used between the pair of devices.

12.5.2.2.1 Data Encryption. Bluetooth devices can perform data encryption using a stream cipher based on Linear Feedback Shift Registers (LFSR). The stream cipher generates a key stream that is used by the sender to encrypt the payload field of each packet using the one-time pad technique (the ciphertext is obtained as a result of the bit-wise exclusive-or operation performed on the payload and the key stream). The recipient of the packets decrypts the encrypted payload field of each packet by generating the key stream and combining it with the encrypted payload fields based on the one-time technique. The stream cipher is initialized by both communicating devices with the device address of the master device, the value of the shared *encryption key,* and the clock of the master device. The stream cipher is resynchronized for each payload using the master's clock. A new key stream is thus generated to encrypt each payload.

12.5.2.2.2 Security Evaluation. The Bluetooth security architecture suffers from some weaknesses in the key management scheme. The main concern is the weakness of the key-generation process for the initialization key. The initialization key is derived from a random number and a secret PIN, whereby the only secret is the PIN. Due to limited capability of human memory, the PIN typically is chosen as a number with at most six digits. A six-digit secret can easily be retrieved by exhaustive search. Another exposure exists when a device's unit key is used as the link key. If a device's unit key is used as the link key for the purpose of parallel or subsequent communications between this device and several other devices, the secret unit key of the device is disseminated to several devices that might include potential intruders. Various types of attacks ranging from the impersonation of the legitimate owner of the unit key to the decryption of encrypted traffic by intruders become feasible based on the knowledge of a device's unit key by intruders.

12.5.3 Relevance of Security Mechanisms in the Data-Link Layer

Although the relevance of security mechanisms implemented in the data link layer is often argued, this question deserves careful analysis in the light of requirements raised by the two different environments in which these mechanisms can be deployed:

1. Wireless extension of a wired infrastructure as the original target of 802.11 and Bluetooth security mechanisms
2. Wireless ad hoc networks with no infrastructure

In (1), the main requirement for data-link-layer security mechanisms is the need to cope with the lack of physical security on the wireless segments of the communication infrastructure. Data-link-layer security is then perfectly justified as a means of building a "wired equivalent" security as stated by the objectives of WEP. Data-link-layer mechanisms like the ones provided by 802.11 and Bluetooth basically serve as access control and privacy enhancements to cope with the vulnerabilities of radio communication links. However, data-link-layer security performed at each hop cannot meet the end-to-end security requirements of applications either on wireless links protected by 802.11 or Bluetooth nor on physically protected wired links.

In case of wireless ad hoc networks as defined in (2), there are two possible scenarios:

- Managed environments whereby the nodes of the ad hoc network are controlled by an organization and can thus be trusted based on authentication
- Open environments with no a priori organization among network nodes

The managed environment raises requirements similar to ones of (1). Data-link-layer security is justified in this case by the need to establish a trusted infrastructure based on logical security means. If the integrity of higher-layer functions implemented by the nodes of a managed environment can be assured (i.e., using tamperproof hardware) then data-link-layer security can even cover higher-level security requirements raised by the routing protocol or the applications.

Open environments, on the other hand, offer no trust among the nodes and across communication layers. In this case, trust in higher layers like routing or application protocols cannot be based on data-link-layer security mechanisms. The only relevant use of the latter appears to be ad hoc routing security proposals whereby the data-link-layer security can provide node-to-node authentication and data integrity as required by the routing layer. Moreover, the main impediment to the deployment of existing data-link-layer-security solutions (802.11 and Bluetooth) would be the lack of support for automated key management that is mandatory in open environments where manual key installation is not suitable.

12.6 CONCLUSION

The need for security mechanisms that cope with the threats that are specific to the ad hoc environment has recently gained attention among the research community. In order to avoid the same problems that arose in wired networks like the Internet, security has to be taken into account at the early stages of the design of basic networking mechanisms like the data-link layer and the network-layer protocols. Since the correct network operation can be heavily jeopardized by threats that range from simple lack of cooperation to routing message modification, ad hoc networks without a proper defense against attacks that are specific to this new networking paradigm cannot exist.

Current efforts carried out by the research community in order to support ad hoc networks with practical security mechanisms have to cope with a challenging environment, where limitations on battery life, computational power, and storage resources, not to mention the lack of any fixed infrastructure, make the design of a security infrastructure very difficult.

The security mechanisms presented in this chapter are a practical response to specific problems that arise at a particular layer of the network stack. However, the proposed solutions only cover a subset of all possible threats and are difficult to integrate with each other. As an example, the secure routing protocols analyzed in this chapter do not cope with the lack of cooperation of the nodes of the network, and are not designed to incorporate a cooperation enforcement mechanism. An exhaustive security infrastructure has to consider a wide range of attacks and has to be made of easy-to-integrate components. Furthermore, security needs may vary according to different networking scenarios and the security mechanisms adopted to combat misbehaving or compromised nodes have to be flexible enough to be used in different environments. The direction that has been taken by the research community in order to support ad hoc networks with security mechanisms confirms this vision, and the proposed solutions are gradually reaching a mature stage, making ad hoc networking a realistic alternative to wireless and 3G networks.

REFERENCES

1. P. Papadimitratos and Z. Haas, “Secure Routing for Mobile Ad Hoc Networks,” in *Proceedings of CNDS 2002.*
2. Y-C Hu, A. Perrig and D. B. Johnson, “Ariadne: A secure On-Demand Routing Protocol for Ad Hoc Networks,” in *Proceedings of MOBICOM 2002.*
3. A. Perrig, R. Canetti, D. Song, and J. D. Tygar, “Efficient and Secure Source Authentication for Multicast,” in *Proceedings of NDSS 2001.*
4. A. Perrig, R. Canetti, J. D. Tygar, and D. Song, “Efficient Authentication and Signing of Multicast Streams over Lossy Channels,” in *Proceedings of IEEE Symposium on Security and Privacy,* 2000.
5. B. Dahill, B. N. Levine, E. Royer, and C. Shields, “ARAN: A secure Routing Protocol for Ad Hoc Networks,” UMass Tech Report 02-32, 2002.
6. Y-C Hu, D. B. Johnson, and A. Perrig, “SEAD: Secure Efficient Distance Vector Routing for Mobile Wireless Ad Hoc Networks,” in *Proceedings of 4th IEEE Workshop on Mobile Computing Systems and Applications.*
7. C. E. Perkins and P. Bhagwat, “Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers,” in *Proceedings of SIGCOMM 1994.*
8. J. Broch, D. A. Maltz, D. B. Johnson, Y-C Hu, and J. G. Jetcheva, “A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols,” in *Proceedings of MOBICOM 1998.*
9. P. Johansson, T. Larsson, N. Hedman, B. Mielczarek, and M. Degermark, “Scenario-based Performance Analysis of Routing Protocols for Mobile Ad Hoc Networks,” in *Proceedings of MOBICOM 1999.*
10. A. Perrig, Y-C Hu, and D. B. Johnson, “Wormhole Protection in Wireless Ad Hoc Networks,” Technical Report TR01-384, Dept. of Computer Science, Rice University.

11. P. Michiardi and R. Molva, “Simulation-based Analysis of Security Exposures in Mobile Ad Hoc Networks,” in *Proceedings of European Wireless Conference, 2002.*
12. D. B. Johnson and D. A. Maltz, “Dynamic Source Routing,” in *Ad Hoc Wireless Networks, Mobile Computing,* T. Imielinski and H. Korth (Eds.), Chapter 5, pp. 153–181, Kluwer Academic Publishers, 1996.
13. C. Perkins, “Ad hoc On Demand Distance Vector (AODV) Routing,” Internet draft, draft-ietf-manet-aodv-00.txt.
14. L. Buttyan and J.-P. Hubaux, “Nuglets: A Virtual Currency to Stimulate Cooperation in Self-Organized Ad Hoc Networks,” Technical Report DSC/2001/001, Swiss Federal Institute of Technology, Lausanne, 2001.
15. S. Buchegger and J.-Y. Le Boudec, “Nodes Bearing Grudges: Towards Routing Security, Fairness, and Robustness in Mobile Ad Hoc Networks,” in *Proceedings of the 10th Euromicro Workshop on Parallel, Distributed and Network-based Processing.*
16. S. Buchegger and J.-Y. Le Boudec, “Performance Analysis of the CONFIDANT Protocol,” in *Proceedings of MobiHoc 2002.*
17. S. Marti, T. Giuli, K. Lai, and M. Baker, “Mitigating Routing Misbehavior in Mobile Ad Hoc Networks,” in *Proceedings of MOBICOM 2000.*
18. P. Michiardi and R. Molva, “Core: A COllaborative REputation mechanism to Enforce Node Cooperation in Mobile Ad Hoc Networks,” in *Proceedings of IFIP Communication and Multimedia Security Conference 2002.*
19. P. Michiardi and R. Molva, “Game Theoretic Analysis of Security in Mobile Ad Hoc Networks,” Institut Eurecom Research Report RR-02-070, April 2002.
20. H. Yang, X. Meng, and S. Lu, “Self-Organized Network-Layer Security in Mobile Ad Hoc Networks.”
21. S. Capkun, L. Buttyan, and J-P Hubaux, “Self-Organized Public-Key Management for Mobile Ad Hoc Networks,” in *Proceedings of ACM International Workshop on Wireless Security, WiSe 2002.*
22. P. Zimmermann, *The Official PGP User’s Guide,* MIT Press, 1995.
23. M. Reiter and S. Stybblebine, “Authentication Metric Analysis and Design,” *ACM Transactions on Information and System Security,* 1999.
24. H. Luo and S. Lu, “Ubiquitous and Robust Authentication Services for Ad Hoc Wireless Networks,” UCLA-CSD-TR-200030.
25. A. Shamir, “How to Share a Secret,” *Communications of ACM,* 1979.
26. IEEE 802.11b-1999 Supplement to 802.11-1999,Wireless LAN MAC and PHY specifications: Higher speed Physical Layer (PHY) extension in the 2.4 GHz band.
27. “Specification of the Bluetooth System,” Bluetooth Special Interest Group, Version 1.1, February 22, 2001, http://www.bluetooth.com/pdf/Bluetooth_11_Specifications_Book.pdf.
28. B. Schneier, *Applied Cryptography,* Wiley, 1996.
29. Stubblefield, Loannidis, and Rubin, “Using the Fluhrer, Mantin, and Shamir Attack to Break WEP,” AT&T Labs Technical Report, 2001.

CHAPTER 13

SELF-ORGANIZED AND COOPERATIVE AD HOC NETWORKING

SILVIA GIORDANO and ALESSANDRO URPI

13.1 INTRODUCTION

Traditional approaches to ad hoc networks do not exploit certain characteristic of these networks (e.g., cooperation and relationships among nodes). A different approach to the network layer is based on the social behavior of people in real life.

A community is the structure that derives from individuals' interactions in a shared environment. The resulting set of interrelated community members is generally called the social network of a community. A fundamental characteristic of communities of individuals is that, regardless of the number of individuals, a single individual needs only little information about other individuals to still be able to potentially interact with a large number (or all) of the community members. An interesting model for social relationships is represented by the small-world graph [33]. The six degrees of separation property illustrates this in the case of human communities. Moreover, communities are often characterized by a highly self-organizing behavior. Insect collectives such as ants or bees and also physical and chemical systems composed of large numbers of individuals or particles interact locally and thereby contribute to global organization, optimization, and adaptation to the environment. Computer networks or distributed systems in general may be regarded as communities similar to the above examples [32].

In particular, a mobile ad hoc network is a community of users and, very similarly to human communities, can be modeled as a small-world graph. Nodes maintain routes not only to nodes discovered by means of the routing protocol (short connections), but also to some other nodes (long connections) that are considered interesting to them. The main result is a self-organizing network that is more naturally suitable and effective for construct-

Mobile Ad Hoc Networking. Edited by Basagni, Conti, Giordano, and Stojmenovic.
ISBN 0-471-37313-3 © 2004 Institute of Electrical and Electronics Engineers, Inc.

ing a citizens' network, which can reduce communication costs and complexity and improve people's ability to share information anywhere and anytime. The self-organization characteristic is one of the fundamental and more innovative aspects of mobile ad hoc networks. Learning from biology, self-organization can be defined as a property of certain dynamical mechanisms whereby structures, patterns, and decisions appear at the global level of a system based on interaction among low-level components. The rules specifying interactions among the systems' components are executed on the basis of purely local information, without reference to the global pattern. A large amount of research in ad hoc routing deals with the networking aspects of these networks. However, traditional schemes for ad hoc networks do not exploit the characteristics more related to self-organization (e.g. cooperation and relationship among nodes).

The Internet Engineering Task Force (IETF) working group on Mobile Ad-hoc NETworks (MANETs) is standardizing routing in ad-hoc networks. The group studies routing specifications, with the goal of supporting networks scaling up to hundreds of routers [1] The work done in the MANET working group relies on other existing IETF standards such as mobile-IP and IP addressing.

A mobile ad-hoc network that explicitly supports the Internet still presents the major salient characteristics we described above, as described in [10]:

1. Dynamic topologies
2. Bandwidth-constrained and variable-capacity links
3. Energy-constrained operation
4. Limited physical security

As presented in [11] and [20], the approach followed by the protocols produced by the MANET working group, and by similar protocols worked on outside of MANET working group is based on the traditional two-level hierarchy routing architecture.

With this view, these routing protocols fall into the class of interior gateway protocols, that is, protocols used to route within a (mobile wireless) network or a set of interconnected (mobile wireless) networks under the same administration authority. However, the managing of IP addresses of the nodes in a mobile ad hoc network is not straightforward. This approach does not fully exploit the cooperation and self-organization peculiarities of ad hoc networks. Therefore, there is a need for another approach, as we explain later.

As discussed in Chapter [10], most of the research on ad hoc networking, in the past and in the recent years, has focussed on this approach. The current chapter illustrates how ad hoc networks can be based on the social model of human communities in order to achieve collaboration and communication, as happens in the real life. The communication task is identified with the capability of delivering some information to a destination. In this case, the critical aspect is the selection of a path from the source to the destination. In Section 13.2, we discuss how this routing task can be performed in a way that reproduces human behavior in a human community. Node collaboration involves several aspects, and, even in this case, it will be addressed with a model that tries to emulate the real behavior of people. Game theory, as we discuss in Section 13.3, considers the cooperation aspects in dynamical system such as ad hoc networks. Games rules model the freedom of every node to choose cooperation or isolation, and, given that players (i.e., nodes) choose rules that maximize their personal benefits, it is possible to address situations of heterogeneity. Finally, as rules and strategies lead to the creation of new network topologies, game theory is the natural tool for studying how the different ways of cooperating affect the evolution of the system.

13.2 COMMUNICATION IN A SELF-ORGANIZED NETWORK

Stanley Milgram introduced the small-world phenomenon with experimental study in social science in the 1960's [25]. These experiments showed that the acquaintanceship graph connecting the entire human population has a diameter of six or less, hence the "six-degrees of separation." The small-world phenomenon formalizes this anecdotal notion by introducing a class of graphs—small-world graphs (SWGs)—that appear to embody the defining characteristics of the small-world phenomenon: very large graphs that tend to be sparse, clustered, and have a small diameter. The behavior of small-world systems is very far from the typical behavior of physical systems. In particular, the notion of being "close," which in physical systems obeys the triangle inequality, it is very different. The fact that node A is well acquainted with B and C, could easily be associated with the situation in which B and C are not even remotely familiar with each other. The main consequence is that, in SWGs, local actions can have global consequences.

The applications of small-world research both encompass and stretch well beyond sociological problems [25] and can include the mobile ad hoc networks. With this view, nodes are in relationships. They need not be physically close, but if they know how to deliver packets to each other, they may violate the transitivity of distances rule of physical systems. That is, while connectivity is achieved through local wireless links (see a well-known model in [27]), the communication is based on a relationship that does not respect the local connectivity. In this way, the communication network can be modeled as a small-world graph. In line with this approach, self-organized routing distinguishes itself from traditional mobile ad hoc routing protocols based, as previously stated, on the traditional Internet two-level hierarchy routing architecture, by emphasizing the self-organizing peculiarities [3]:

1. Self-organized networks are nonauthority-based networks; that is, they can act in independently of any provider or common denominator, such as the Internet. Therefore, whereas authorities or special role nodes do not belong to this architecture, it is mandatory to regulate the way the nodes cooperate, as explained in the next sections.
2. Self-organized networks are not regularly distributed, neither in terms of node (network density), nor in terms of topology. Potentially, they can be very large.
3. Self-organized networks are based on the cooperation between nodes. Thus, networking functions must be based on the support of other nodes (aided routing; see [16]).

13.2.1 Terminode Routing

One way to achieve self-organization is terminode routing [4]. Terminode routing is a self-organized routing scheme where by internode communication, even in a large network, relies solely on local information and interaction with other nodes. The routing task is accomplished by two interacting protocols: Terminode Local Routing (TLR) and Terminode Remote Routing (TRR). The TLR mechanism is used to reach destinations in the vicinity of a node and uses local routing tables that every node proactively maintains for its close neighbor nodes. It is very similar to the traditional mobile ad hoc routing protocols presented in Chapter 10 and, therefore, not a main point of interest for our analysis.

More interesting is the approach taken whenever a source has to send data to a remote destination. The TRR protocol works with geographic information, that is, node coordinates in two- or three-dimensional space. Geographic (or position-based) routing [15] allows nodes in the network to be nearly stateless; the routing based is on information maintained locally. For that reason, it is more scalable than traditional routing; the information that nodes in the network have to maintain is not much more than that of their neighborhood.

However, greedy geographic routing could fail in some scenarios, so the TRR protocol introduces two novel elements: the *friend nodes* and the *anchor paths*. These two elements allow for a more valid use of geographical routing by structuring the communication network as a SWG.

Friends Nodes. In order to achieve the SWG properties, the authors introduce nodes with a special role: the friends. Similarly to social networks, the mobile ad hoc network is seen as a large graph, with edges representing the "friend relationship." As explained in [4], B is a friend of A if (1) A thinks that it has a good path to B and (2) A decides to keep B in its list of friends. A may have a good path to B because A can reach B by applying TLR, or by geodesic packet forwarding, or because A managed to maintain one or several anchor paths to B that work well [4]. The value of a path is given in terms of congestion feedback information such as packet loss and delay. The way a node gets in touch with its potential friends is thought to be similar to the human behavior, such that the friend relationship works similarly to a relationship in a human community and the networks presents SWG properties. More precisely, a node, being used by some person and running certain applications, has a given number of contacts with other nodes, which potentially are remote. Additionally, by its nature the friend relationship is present within the node's neighborhood, as neighbor nodes are known and reachable.

With this approach, a node is assumed to select its friends from a list that includes some remote nodes in addition to the close nodes of the neighborhood. Therefore, this set is suitable for constructing the short- and long-range connections of a SWG. And each couple of nodes are likely to be connected through a short sequence of intermediate vertices. This means that any two nodes are likely to be connected to a small number of intermediate friends. The key element of the terminode routing scheme for achieving the SWG characteristic is the way the friends are selected (and maintained). The characteristic that is reproduced in the scheme is the presence of a small fraction of long-range edges, which connect otherwise distant parts of the graph, while most edges remain local, thus contributing to the high clustering property of the graph.

Anchor Paths. In order to exploit the potential of this approach, in conjunction with the geographical routing, TRR introduce an additional novel element: the anchor path. In contrast to traditional routing paths, an anchor path does not consist of a list of nodes that a packet has to visit before reaching the destination. An anchor path is a list of fixed geographic points called anchors. If we make a comparison with traditional paths, the advantage of using geographic points is given by the fact that they do not move. So, whereas traditional paths can become invalid when the nodes move, the anchor paths, once established, can always be used, even in a mobile environment.

The friends are used to discover the anchor paths. When a source S wants to discover a path to destination D, it requests assistance from some friend. If this friend is in condition to collaborate, it tries to provide S with some path to D (it can have it already or try to find it, perhaps with the collaboration of its own friends) [4]. Therefore, an anchor path results in a list of positions of friends. Now, given that the friend relationship can represent, with some probability, a long-range connection, the anchor path is likely to be much shorter than the number of hops necessary between the source and the destination.

Example. Suppose that a source S wants to communicate with a destination D. Traditional routing methods try to discover a path from S to D. In Figure 13.1, we use as an example the shortest path (in terms of number of hops) from S to D, and we represent it with dotted lines. It consists of six hops.

With the SWG approach of terminode routing, S contacts its friend F1 to get a path to D, and F1 contacts F2, which knows a path to D (see [4] for technical details). In this case, the resulting anchor path (S, AP1, AP2, D), illustrated by thick lines, consists of just three hops and shows that this communication network has a smaller characteristic path length compared to the traditional routing approaches.

13.2.2 Grid Location Service (GLS)

Another example based on the small-world phenomenon is presented in GRID [19]. In this work, the authors build a structure that can be seen as a small-world graph for the location aspects. At the routing level, GRID uses geographical forwarding to take advantage of the similarity between physical and network proximity. In order to support geographical forwarding, the GRID scheme introduces the Grid Location Server (GLS). GLS is based on the idea that a node maintains its current location in a number of location servers distributed throughout the network. These location servers are not specially designated; each node acts as a location server on behalf of some other nodes. The location servers for a node are relatively dense near the node but sparse farther from node; this ensures that any node near a destination can use a nearby location server to find the destination, and also limits the number of location servers for each node. Therefore, the graph representing the connection between each node and its location

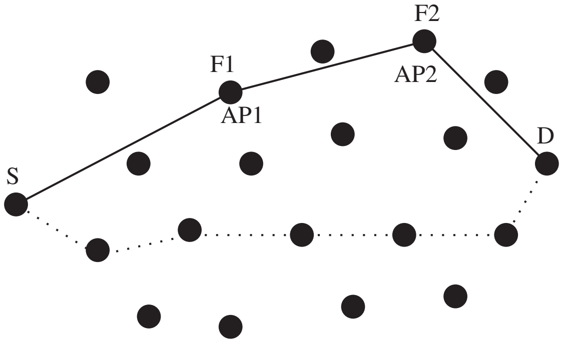


Figure 13.1. The communication network when traditional routing (dotted lines) and anchor paths (thick lines) are used.

servers has the sparse, clustered, and small-diameter characteristics of a small-world graph.

13.3 COOPERATION MODELING

In all the communities, a certain degree of cooperation is needed to carry on all the common tasks. Cooperation can be the result of a strong hierarchy induced in the group (e.g., in ant or bee colonies, where each member has a clear role and there is a set of critical tasks is assigned to each role), or of a more or less spontaneous altruism (like the mutual grooming of birds). In both cases, cooperation leads to undoubted benefits: group equilibrium, probably reached after a long time, is maintained, and every community member achieves some personal advantage by helping others.

13.3.1 From Closed Ad Hoc Networks to Spontaneous Networking

Ad hoc networks were first developed for military use. The current trend is to move from these ad hoc networks managed by some authority toward a more spontaneous networking ([18]). This progression from closed small networks, managed by a single authority (i.e., a military unit or a civil protection team), to open networks of reasonable size that spontaneously work when two or more people want to exchange data for some reason, will change many rules. In fact, whereas in closed scenarios cooperation is not an issue, since all the involved devices have the same purpose and belong to the same "community," in future ad hoc networks it is fundamental to understand whether a user, who in principle just wants to have a service, should work to have it. Or even better: if a user enters into an already working ad hoc network, why should he offer help, if it is not necessary? Moreover, cooperation does not come for free: processing and forwarding someone else's traffic can be extremely costly in terms of spent energy, probably the most precious resource in such a system. A node can be for this reason encouraged to be *selfish,* that is, to be part of the network without helping other nodes (Chapter 12).

Recent works [21, 24] have shown the severe impact that selfish nodes can have on network performance: as the percentage of selfish nodes increases, the network throughput decreases almost linearly, causing serious problems.

Moreover, traditional tools used to model ad hoc networks seem not very able to model a high-level property like cooperation. It is also difficult to study the correlations of a collaborative behavior with spent energy and received service. For this reason, it is quite natural to look at disciplines like economics and the social sciences in order to find models and analytical tools to borrow. Here we present the theoretical aspects of this problem; the systems and protocols aspects are presented in Chapter [12].

13.3.2 Game Theoretical Model(s)

During the last fifty years, researchers were able to describe in great detail human society at many levels, from an individual point of view, to economic entities, up to national interactions. At every level, they have been able to predict behaviors and to choose optimal strategies at critical moments, or have failed for lack of information or wrong models, which has led to several wars. Such systems are composed of a huge number of complex entities, each concerned with their personal status, and each with limitations and unpredictable behavior. A very useful tool that has been extensively used, developed, and refined during this time is *game theory*. If the reader is not familiar with basic game theoretical concepts, the simpler notions are presented in Section 13.5, but we will try to make the remainder of the section not too technical and understandable without any knowledge of the subject. Paragraphs entitled "Formal Explanation" can be ignored if the reader is not interested in technical details.

Basic Model. In this field, the research has mainly been concentrated in finding mechanisms or protocols for stimulating cooperation and to study the behavior of the proposed mechanism. Models based on game theory have not been explored much in the ad hoc networks literature. There exist some works [6, 9, 22, 30] introducing strategies for cooperation in ad hoc networks that implicitly are based on a game theoretic model. However, these models are usually not formally stated, and for this reason the current trend is to look for a general, unifying model [30, 31]. In all the proposed models, the only common background is that, at a suitable level of abstraction, nodes composing an ad hoc network can be seen as interacting entities that can request or offer a service. A single node can be, and usually is, both a user and a provider. For the sake of clarity, we will just consider the packet forwarding functionality. It is possible, at least in principle, to extend all the considerations we will make to a large class of other services that need global cooperation, like routing, even if, in practice, it can be difficult to deal with all the services at the same level. The time is discrete (divided in slots), and in every round one or more nodes ask some other node to forward messages for them. A potential provider can accept the task or ignore it. In the latter case, ignored packets are lost, and the sources have to fix the problem in some way. Given the infrastructureless nature of an ad hoc network, nothing can work if no providers accept relay requests. On the other hand, each node is really concerned with energy consumption, and would prefer that someone else carry out the task, since there is no explicit payment for it.

Formal Explanation. This situation is reminiscent of the prisoner's dilemma: in a single round, with no future play perspective, nodes have to decide whether to accept or reject any forwarding request from their neighbors. If they all accept, than the system payoff is maximized; but from a personal point of view, it would be better not to accept and let some other node forward that packet. On the other hand, if a node is going to accept a forwarding request, but sees its forwarding request rejected, then it is in deep trouble, whereas if no node accepts, the the payoff is very low both from a system point of view and from a personal one, but it is always better than being the only node working. The payoff matrix for a two-node network is the same as that presented in Section 13.5, with moves labeled as *Acc* (accept to forward) and *Rej* (reject forward requests) and with $a > b > c > d$:

	Acc	Rej
Acc	*b, b*	*d, a*
Rej	*a, d*	*c, c*

Packet forwarding in a network can be seen as an infinite repetition of such a game, with forwarding requests made at every round and decisions taken at the beginning in the form of a *strategy* (e.g., always accept, always reject, accept to forward just in even rounds, do to others what they are doing to you, do to others what you expect they are going to do to you, and so on).

System Equilibria. Once a model in terms of a game is precisely given, equilibria of the system can be studied. These represent the set of stable network states in which nodes will arrive driven by their personal interests, which usually do not match with a global optimization, and from which they cannot leave by changing behavior in a unilateral way.

However, if in a repeated game a set of nodes agrees on some desired and specific equilibrium (since, generally, there are many of them), it is possible to force all the nodes to behave in that way by punishing the ones that deviate, thus making other strategies less productive than the chosen one.

Formal Explanation. As has very well studied in prisoner's dilemma, the unique Nash equilibrium is the noncollaborative action that would imply a nonworking network in our case. However, since we are in the presence of repeated plays, if a certain percentage of nodes agrees on a collaborative behavior, then it is possible to choose a strategy that makes collaboration advantageous for every node. It will be enough to "punish" nodes that decided not to cooperate by excluding them from the network for a sufficient time, that is, a sufficient number of rounds to make payoff obtained after a defection less than losses given to network exclusion. If the future is important enough, then well-known strategies like TIT FOR TAT are optimal in this sense [2].

In addition to these simple considerations, many approaches have been taken in recent times, arriving at different solutions that stress different aspects, and that are, in many ways, incomparable. We present here some approaches, trying to understand what they have added to the basic model, and what are their possible weak points.

A First Model: Rewarding Selfishness. In [21], which is the starting point for many other works, Marti et al. are not concerned about cooperation itself, but about offered throughput in the network, which is, of course, affected by selfishness. They propose to equip every node with a *watchdog,* a unit that listens to all the communications that arrive at the node itself. It is possible, in many cases, to listen to service request messages, and to understand whether the requested node does its task or not, if the nodes are near enough. If some node is not cooperating, it will not be asked to do anything, but it will be still able to use network services. The solution increases the performance of the network in terms of percentage of delivered packets, since less paths containing bad nodes are used, but it also encourages selfishness, as long as a sufficient number of nodes ensure network connectivity. Moreover, although the watchdog unit is a nice theoretical departure from the local knowledge of network operations,[1] it has been used with very strong hypotheses, like equal communication range for all the nodes, which is very unlikely to happen in practice.

Formal Explanation. The watchdog unit is used to directly observe the moves played by neighbor nodes (other players do not affect the payoff of a node). During a given time slot

[1] One of the hypotheses in game theory is that moves are simultaneous and observable by other players, which, of course, does not happen in ad hoc networks.

t_k, node i observes a certain fraction of packets that its neighbor node j has to forward [$R_j^i(t_k)$] and the packets that it effectively forwards [$F_j^i(T_k)$]. If $F_j^i(T_k) \ll R_j^i(t_k)$, then, assuming a perfectly working watchdog, node j is probably[2] not forwarding traffic, and node i can record this information, which is passed to a path-rater component. Every time i is asked to find a path to some node z, it will not use routes containing nodes marked as "bad" by its path-rater units, assuring higher probability that delivered packets will pass on the demanded route. Coming back to the model, Marti et al. propose this strategy: "Always cooperate, and if someone is not doing the same, do not use him, since he is unreliable," which is, of course, very far from any equilibrium (since one who deviates is rewarded instead of being punished). In terms of performance, the solution works, because throughput tends to increase.

CONFIDANT: An Evolutionary Model. Buchegger and Le Boudec [6, 7] start from the possibility of exploiting the previous solution, adopting an *evolutionary* approach: they see nodes as an interacting population, and look for a strategy that yields more benefit than any other strategy that a newcomer node can adopt [12]. If nodes in a network all adopt such a mechanism (as if it were coded in their genes), then a node using a different strategy (a mutation, because it arrived from another population in which evolution led to other solutions) could not "survive," that is, receive more service than preexisting ones, and should change its strategy (adopting the official one), or die, being excluded from network use. In [2] it is shown that a strategy can be evolutionary successful only if it is adaptive. For this reason, nodes have to be equipped with a watchdog unit, to observe the behavior of neighbors and adapt to their behavior. Moreover, they are also organized in a friendship network (see Section 13.2). When a node discovers that one of its neighbors is not cooperating, it starts warning all its friends, which mark the misbehaving node as bad. At this point, the selfish node is cut off from network services, because bad nodes are not served. Warning messages are surely the main weak point of the first proposed solution [6]: they add overhead to the network, and it is possible to spread fake information, leaving an open door to denial of service attacks. For this reason, in [7] the authors extended the protocol in order to trust messages about unknown nodes only when they arrive close in time and from a large number of friends.

Formal Explanation. The idea of this work is to find an evolutionary equilibrium in the repeated game. Such a strategy would guarantee stability over time and the failure of any other strategy, when adopted by small enough clusters of newcomer players. Again, every node i can observe $R^i{}_j(t_k)$ and $F^i{}_j(t_k)$, and, when $R^i{}_j(t_k) \ll R^i{}_j(t_k)$, punish node j by not serving its requests. Moreover, to emulate the instantaneous knowledge of a played move by all the player, alert messages are sent to some trusted friend, which in turn will stop helping node j. The strategy is an approximation of the TIT FOR TAT one: start helping, and treat nodes in the same way they treat you. This strategy undoubtedly has good properties (first of all: it is extremely simple), but it permit noncollaborative nodes, after one turn of punishment, to be reintegrated into the network if they start helping. In the cited works, however, there is no formal analysis of the model used.

CORE: A Reputation-Based Model. The solution proposed in [22, 23] by Michiardi and Molva overcomes the problem of alert messages, and makes it explicitly possible to

[2] It is not possible to deduce sure facts by pure observation, as the authors point out with some examples.

end a punishment when a node starts behaving well. For this reason, every node has a local knowledge of the reputation of other nodes, which it can modify in just two cases:

1. With local observation (again, with a watchdog unit), it can increase or decrease other nodes' reputation, depending on how are they behaving. In this way, a previously noncollaborative node can start helping and be reintegrated into the network, even if very slowly.
2. With indirect deductions that can just be positive. It is necessary to have, after the execution of every cooperative functionality, a replay message containing the names of every node that participated (that can be considered good). Nothing can be deduced if a node is not in a replay message.

The authors analyze their solution in game theoretic terms, proving that if only half of the nodes adopt it, the remaining nodes have to collaborate in order to use the network. In principle, the model is made heavy by the presence of indirect reputation. It is not essential to collect informations about nodes not in the neighborhood, whereas from a practical point of view, in presence of mobility, it can be useful to have some data about a newcomer node, either positive or uncertain. Moreover, the computation of indirect reputation in some cases can be quite heavy in computational terms. For example, the authors claim that when a route is established with the DSR protocol, it is possible, to a certain extent, to infer that all the nodes in the path cooperated to build it.[3] This means that at every route establishment, the returned message (a list of nodes) has to be analyzed. However, this is a theoretical weak point, which, in practice, can be disregarded.

Formal Explanation. The analyzed papers were written at an embryonic state of the model, so many observations we are making at this point could have changed by the time of this writing. The base model is always the same game, but the authors do not try to have instantaneous global information of the move played, letting nodes know everything about their neighbors, and something about past moves of distant nodes, if they played collaboratively. For this reason, if a player is not cooperating, it is punished by all its neighbors, which block traffic from and to it, which is enough to make selfishness unattractive. If the model is not precisely stated, the authors use ERC theory [5] to model a multiplayer prisoner's dilemma with a fair share of resources. ERC perfectly explains weird behaviors exhibited by humans when playing games during research experiments. The observed tendency to depart from Nash equilibria is explained by a major satisfaction deriving not only from monetary payoff, but also from the distance of others' payoffs. Ad hoc nodes probably are selfish in an absolute way, and it is difficult to compare them with humans, so a classical game theoretic analysis would have been more appropriate.

Nuglets: A Market Model. A radically different solution was proposed in [8, 9] by Buttyan and Hubaux, who model a network as a market in which services are exchanged. A virtual economy, based on a virtual currency called a nuglet (or bean), is then introduced, forcing nodes to pay to have their packets forwarded, and to be paid when they forward some data. In this way, a selfish node would soon use up its nuglets, and would be forced to cooperate in order to send other packets. For the first time, energy considera-

[3] This is true if the routing protocol is robust against changes of established routes, that is, if it is not possible for a node to insert itself in a route in which it is not present.

tions are explicitly made, pointing out that the first goal of nodes is to survive, finding the optimal number of messages to send and forward, given the battery power of every node. It is clear that nuglets should be managed in a tamper-proof part of the node, ensuring that a malicious user does not change its device in order to steal, forge, or throw away nuglets. Some of these problems have been solved in [9] with the introduction of counters in place of nuglets, always managed in a separate hardware module.

Moreover, the proposed solution has been enriched by Zhong et al. in [34], where they propose a solution for which no tamper-proof hardware is needed, but a centralized server acting as a bank must exist, if not within the ad hoc network, then at least in another network reachable by all the nodes.

Formal Explanation. This is one of the very first solutions proposed in the literature and, interestingly enough, it is not watchdog-based, the main weak point of all the analyzed mechanisms. Every node has an initial account of C nuglets and a battery capability of B. Packets are generated at a constant average rate f_o, and every node receives packets to forward at rate f_r. It is possible then to compute the number of packets to send and forward in order to maximize the throughput while not running out of battery power. The point is clearly a fair optimum, since all the nodes are forced to spend the same amount of energy for forwarding and for sending their data, but has to be enforced in hardware with a tamper-proof module, since no node can observe what its neighbors are doing. Moreover, if not all the nodes are equipped with such a module, selfish nodes would have an extremely easy life, with self-limiting neighbors that do not punish them. Four different dynamic strategies for managing nuglets are analyzed, showing that the most generous one (forward all the packets until you reach your limit) is also the best-performing one. Again, no formal analysis of the underlying model has been done.

GTFT: An Energy-Aware Model. Srinivasan et al. [30] propose a trade-off between the existent solutions. Since energy is the main concern of authors, nodes are seen as partitioned into energy classes (depending on their energy constraints), and communications are arranged in sessions, each with an associated energy class, determined by the "weaker" node in the chain of forwarders. Each node, before the start a session, asks to all the nodes in the route that will be used if they are going to support the session itself. It is possible to compute optimal strategies for every class having complete off-line network information, and the authors also propose an algorithm that permits nodes to reach optimal allocations (by knowing them in advance) by simply recording how they are treated by other nodes, and keeping things in balance, substituting nuglets with explicit requests to participate in sessions. The solution does not enforce cooperation in the sense that other works do, since a node is never really cut off from network services, and can continue to communicate without forwarding.

Formal Explanation. Every node i is associated with an energy class E_n, and has to decide the ratio τ_{im} of sessions of type E_m to accept, where E_m is the energy class of the node with less energy (in absolute terms) taking part to the session. If ratios are stationary, that is, if they do not vary during the time, it is possible to compute the Pareto efficient point by solving a system of optimization problems, in which users want to maximize throughput, given that the battery lasts enough (for example, an entire day). These problems involve much off-line information about the network, like the probability that sessions contain a certain number of nodes and the number of nodes in every energy class. Since this is practically unfeasible, a mechanism based on TIT FOR TAT is also presented. Every node records how many sessions of type E_k are accepted (with respect to all the requests for that type), and how many sessions of the same type, initiated by it, were accepted (always in percentage). The GTFT (Generous TIT FOR TAT) strategy consists in keeping the two values in (not strict) balance with the optimal ratio, previously computed. It is shown that it constitutes a Nash equilibrium of the repeated game (in an elegant way, without formally stating the game), and that it assures the same throughput reached at the Pareto efficient point.

A General Model. The authors of [31] do not present any cooperation-enforcement mechanism, but they give a model in order to understand whether cooperation is really a problem in every case, and if yes, when it is possible to obtain it. In this paper, they propose an approach to cooperation based on Bayesian games, in order to model the uncertainty of nodes about their characteristics, and look for the equilibria points of the system, as well as for the class of enforceable cooperative behaviors.

Each node is then endowed with some information about its neighbors and their actions, which includes its neighborhood, the traffic it sent and has to send, and the traffic it received. Prior to choosing its next action, a node has an opportunity to analyze the past behavior of its neighbors and its priorities in terms of energy consumption and throughput, and to decide, consequently, how to act. In deciding to whom to forward packets and to whom to discard packets, the node trades off the costs (energy consumption), the benefits (network throughput), and the collaboration offered to the network by the neighbors. This implicit incentive causes a neighbor to act in a selfish way only when obliged by energy constraints, but each node tends to cooperate with collaborative nodes.

It is possible to show that the noncooperation is always an equilibrium (even if it does not necessarily imply that this will be the general behavior), and that, in general, there is a limit on the traffic a generic node is willing, at maximum, to forward, thus inducing a bound on network performance. However, this limit depends on the characteristics of the nodes and on the obtained knowledge: the less a node is trusted, the more, in general, it should work to have its packets forwarded. Moreover, the amount of cooperation that can be forced depends on the mobility of nodes and on the network size: in very mobile systems it is difficult to locally punish selfish nodes if they are staying for a short time near damaged nodes.

Formal Explanation. The authors of this paper modeled the life of an ad hoc network as a discounted repeated game, with the discount factor depending on the mobility of the network. They chose an imaginary utility function for nodes that (indirectly) depends on spent energy and on received cooperation, with the importance of each component given by its energy class (as in [30]). Equilibria in the single-shot game depend on the knowledge of others' energy classes, but the noncooperation is always present. Under repetition, the discount factor makes cooperative behavior impossible in every case (by Nash folk theorems), but it is still possible to have "kind" behavior by adopting an evolutionary approach.

Discussion and Future Directions. The common idea behind all the presented works is a model (explicit or implicit) based on classical game theoretical concepts: every player knows everything about the others, about the rules, and about the payoffs. The main concept is that, if node i is rational and has access to information like number of sent packets $S_i(t_k)$, number of forwarded packets $F_i(t_k)$, number of packets it was required to forward $R_i(t_k)$, and battery power $B(t_k)$, where t_k is the kth time slot, then it is enough to study a maximization problem for every node, of the form maximize $S_i(t_k)$, considering as a constraint that packets not forwarded are lost, and do not count in the function to be maximized. It is then possible to find optimal quantities of traffic to be forwarded for each node (at least on average, with appropriate hypotheses on traffic generation and mobility), that enforce a certain degree of cooperation determined by the need for communication of every node (a *do ut des* concept). In [9, 30] an ulterior constraint (extremely important) has been added: the energy spent by every node to send and forward traffic has to be less than the battery power available to the node itself. This consideration, present in the mind of the authors of all the papers cited in this section but often not used in the models, has two interesting implications:

1. Selfishness has a formal explanation: if no energy considerations are made in the model, there is no difference between helping others or not.
2. There is the need for every node to find the best trade-off between throughput and consumed energy, assuming that its neighbors are also doing the same.

We believe that classical game theory can be a limited model for real scenarios in which each node can be seen as an entity that barely knows its neighbors (before they start exchanging traffic, they know just their names), and that after each communication receives some imprecise feedback from the environment. It can monitor neighbors in order to find out what they are doing, but it is not practical to assume a perfect knowledge or synchrony in its actions. Often, it is not possible to know the personal preferences in a given moment, nor the motivations that are pushing other nodes to act like that.

Many works ([17] and [14] for a wired-networks case) have tried to overcome these limitations, with interesting results.

13.4 CONCLUSIONS

We have attempted to justify the claim that the qualitative structure of ad hoc networks is fundamental in determining and studying both their structural and dynamical properties. Ad hoc networks, when their characteristics as communities of nodes are highlighted, express very interesting properties. In particular, it has been shown how this approach influences the communication network and, therefore, the routing, and how this can be beneficial for the node cooperation. However, whereas in the case of routing there are several traditional approaches that work equally well, for cooperation, this approach is essential. Intuitively, game theory exploits the main novel characteristics of an ad hoc network: games rules can be interpreted as the interaction, cooperation, or even competition among nodes. Nodes can follow different strategies, and the effect of them will be reflected in the network evolution. Nodes will try to optimize their own payoff, but as they also need to have a functioning network, they adopt cooperative behavior to obtain better network performance, unless they are selfish or malicious, and, in this case, game theory helps to individuate punishment strategies. We presented a first model that formalizes these properties using game theory. However, the research in this field has just started and all the issues related to how to exploit the social and economic characteristics of ad hoc networks represent interesting topics for further research.

13.5 ELEMENTARY CONCEPTS OF GAME THEORY

We present here a few basic concepts of game theory that will help the reader to better understand the first part of the chapter. Obviously, we do not pretend this to be exhaustive, so the reader is really encouraged to get a deeper view from a book entirely focused on the subject (it is impossible for us to select just one or two of the many books available).

When n rational entities interact for some reason, they can be modeled as players in a *game*. A strategic game has its rules, in terms of moves every player can make, and every player has personal preferences about the results of the game. Formally, the basic blocks of a game[4] are:

- A set N of n players
- A set A_i of moves for every player, defining a global action space:

$$A^* = \times_{i=1}^{n} A_i$$

- A payoff function p_i: $A^* \to \mathbb{Z}$ for every player, defining the global payoff:

$$p^*(a) = \times_{i=1}^{n} p_i(a)$$

Every player knows the rules of the game and rational acts in order to maximize the payoff.

Example 1. Probably the most classical problem is the so called *Prisoner's Dilemma* (see [13] for a history). Two suspects of a heinous crime are caught after a minor infraction. They are kept in separate rooms, and each one has to autonomously decide whether to confess to the big crime or not. If only one of them confesses, he will be freed and used as witness in the trial, whereas the other one will receive a sentence of ten years. If they both confess, they will spend five years in prison, whereas if neither does, they will pay with "just" one year of imprisonment for the minor crime.

This can be seen as game with two players, where each one has two possible moves: confess and do not confess. The payoffs can be the number of years they are sentenced to, but in negative terms (since it is a cost).

A very useful way to represent two-player games is via payoff matrix:

	C	*D*
C	–7, –7	0, –10
D	–10, 0	–1, –1

In the rows (columns) are represented Player 1 (2) moves, and in position (i, j) there are the payoffs when Player 1 chooses her ith move and Player 2 plays his jth move.

Probably the most important concept concerning games is *equilibrium*: a move a^* is an equilibrium point if no player, with a unilateral deviation, can increase her payoff. In for-

[4] We present here a simplified version of strategic games, where user preferences are substituted with payoffs. Although this is not generally equivalent to the general case, in many practical cases there is no difference.

mal terms, $a^* = (a_1^*, \ldots, a_n^*)$ is a pure equilibrium for the game $G = (N, A^*, p^*)$ if (and only if):

$$\forall i \in N . \forall a_i \in A_i . p_i(a^*) \geq p_i(a'^*)$$

where a'^* is the same as a^* except for Player i, who plays a_i instead of a_i^*.

Example 2. The prisoner's dilemma has a unique pure equilibrium, one in which both players confess!

The following game has no equilibrium:

	P	*D*
P	1, –1	–1, 1
D	–1, 1	1, –1

Example 2 shows that the concept of pure equilibrium is too weak, and for this reason it has been generalized to mixed (Nash) equilibrium ([26]). Let G' be a game derived by $G = (n, A^*, p^*)$ in the following way:

- The set of players is the same.
- The moves of player i are all the possible probability distributions over A_i.
- The payoff functions are defined on lotteries over combinations of moves.

It is possible to prove that every game has at least a mixed equilibrium, even if the problem of finding the equilibria points has no efficient solution at this moment.

Many variants of strategic games have been presented in literature; we only report on two basic forms here.

In repeated games, a strategic game is repeatedly played, either a finite or an infinite number of times. The moves of single-shot games are combined into *strategies* (action plans) that can be more or less complex. The payoff is a combination of single-game payoffs, considering that past results are "heavier" than future ones, because there is always the possibility that a game is interrupted, or that two players do not meet again (i.e., there is a *discount factor* on subsequent outcomes). The equilibrium is defined over strategies, and it is possible to prove the following facts:

Proposition 1. If the game is repeated a finite number of times, then the Nash equilibria are sequences of Nash equilibria of the constituent game.

Proposition 2. It is possible to have Nash equilibria in infinitely repeated games that are not sequences of Nash equilibria of the constituent game.

The first proposition is the consequence of "backward induction" reasoning: if there is a last move that will be played, then it must be a Nash equilibrium of the constituent game, then also the move before, and so on until the first one. The second proposition is proven with the so-called Nash folk theorems: if all the players agree on a desired sequence of moves, then it is possible to enforce it by punishing deviating players for a sufficient time.

It is in fact sufficient to make a profit derived from a deviation less than the losses that follow this decision, and a rational player will avoid deviations.

In Bayesian games, every player has a secret (her type) which conditions her payoffs, and has a prior belief of the secrets of other players, that is a distribution for the type of every player. The equilibrium is now defined in terms of lotteries of personal beliefs: every player will chose her best response to the distribution of possible moves of others.

ACKNOWLEDGMENTS

This work was partially funded by the Information Society Technologies Programme of the European Commission, Future and Emerging Technologies, under the IST-2001-38113 MOBILEMAN project.

REFERENCES

1. The internet engineering task force mobile ad-hoc networking page (manet): http://www.ietf.org/html.chrters.manet-charter.html.
2. R. Axelrod, *The Evolution of Cooperation.* Basic Books, New York, 1984.
3. L. Blazevic, L. Buttyan, S. Capkun, S. Giordano, J. P. Hubaux, and J. Y. Le Boudec, “Self-organization in mobile ad-hoc networks: The approach of terminodes,” *IEEE Communications Magazine, 39*(6), June 2001.
4. L. Blazevic, S. Giordano, and J.-Y. Le Boudec, “Self-Organizing Routing,” *Cluster Computing Journal, 5*(2), April 2002.
5. G. E. Bolton and A. Ockenfels, “ERC: A Theory of Equity, Reciprocity and Competition,” *American Economic Review, 90:* 166–193, 2000.
6. S. Buchegger and J.-Y Le Boudec, “Performance Analysis of the CONFIDANT Protocol: Cooperation of Nodes—Fairness in Distributed Ad-Hoc Networks,” in *Proceedings of IEEE/ACM Workshop on Mobile Ad Hoc Networking and Computing (MobiHOC),* Lausanne, Switzerland, June 2002.
7. S. Buchegger and J.-Y. Le Boudec, “The Effect of Rumor Spreading in Reputation Systems for Mobile Ad Hoc Netorks,” in *Proceedings of WiOpt 2003,* pp. 131–140.
8. L. Buttyan and J. P. Hubaux, “Enforcing Service Availability in Mobile Ad-Hoc WANS,” in *Proceedings of IEEE/ACM Workshop on Mobile Ad Hoc Networking and Computing (MobiHOC),* Boston, MA, August 2000.
9. L. Buttyan and J. P. Hubaux, “Stimulating Cooperation in Self-Organizing Mobile Ad Hoc Networks.” *ACM Journal for Mobile Networks (MONET), Special Issue on Mobile Ad Hoc Networking,* 2002.
10. S. Corson and J. Macker, “Mobile Ad Hoc Networking (MANET),” ietf rfc 2501, January 1999.
11. J. Macker and S. Corson, “Mobile Ad Hoc Networks: Routing Technology for Dynamic Wireless Networking,” in *Ad Hoc Networking,* IEEE Press/Wiley, 2003.
12. R. Dawkins, *The Selfish Gene.* Oxford University Press, Oxford, 1976.
13. E. R. Weintraub (Ed.), *Toward a History of Game Theory.* Duke University Press, Durham, NC, 1992.
14. F. Eric and S. Shenker, *Learning and Implementation on the Internet,* 1997.
15. S. Giordano, “Mobile ad hoc networks,” in *Handbook of Wireless Networks and Mobile Computing.* Wiley, New York, 2001

16. S. Giordano and I. Stojmenovic, "Position Based Ad Hoc Routes in Ad Hoc Networks," in *Handbook of Ad Hoc Wireless Networks,* M. Ilyas (Ed.). CRC Press, to appear.
17. E. Kalai and E. Lehrer, "Rational Learning Leads to Nash Equilibrium," *Econometrica, 61*, 5, 1019–1045, September 1993.
18. A. Westerlund, L. Feeney, and B. Ahlgren, "Spontaneous Networking: An Application-Oriented Approach to Ad Hoc Networking." *IEEE Communications Magazine,* June 2001.
19. J. Li and J. Jannotti and D. De Couto and D. Karger and R. Morris, "A Scalable Location Service for Geographic Ad Hoc Routing," in *Proceedings of ACM MOBICOM 2000.*
20. J. P. Macker, V. D. Park, and M. S. Corson, "Mobile and Wireless Internet Services: Putting the Pieces Together," *Communication Magazine,* June 2001.
21. S. Marti, T. J. Giuli, K. Lai, and M. Baker, "Mitigating Routing Misbehavior in Mobile Ad Hoc Networks," in *Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking,* pp. 255–265. ACM Press, 2000.
22. P. Michiardi and R. Molva, "CORE: A Collaborative Reputation Mechanism to Enforce Node Cooperation in Mobile Ad Hoc Networks," in *Proceedings of the Sixth IFIP Conference on Security, Communications, and Multimedia (CMS 2002),* 2002.
23. P. Michiardi and R. Molva, "Game Theoretic Analysis of Security in Mobile Ad Hoc Networks." Technical Report RR-02-070, Institut Eurecom, April 2002.
24. P. Michiardi and R. Molva, "Simulation-Based Analysis of Security Exposures in Mobile Ad Hoc Networks," in *Proceedings of the Mobile Ad Hoc Networks European Wireless Conference,* 2002.
25. S. Milgram, "The Small World Problem, *Psychology Today,* 1967.
26. J. F. Nash, "Equilibrium Points in N-person Games," *Proc. Nat. Acad. Sci. U.S.A., 36,* 48–49, 1950.
27. P. R. Kumar and P. Gupta, "The Capacity of Wireless Networks," *IEEE Transactions on Information Theory,* March 2000.
28. E. M. Belding-Royer, "Routing Approaches in Mobile Ad Hoc Networks," in *Mobile Ad Hoc Networking,* IEEE Press/Wiley, 2004.
29. P. Michiardi and R. Molva, "Ad Hoc Network Security," in *Ad Hoc Networking,* IEEE Press/Wiley, 2003.
30. V. Srinivasan, P. Nuggehalli, C. F. Chiasserini, and R. R. Rao, "Cooperation in Wireless Ad Hoc Networks," in *Proceedings of IEEE Infocom 2003.*
31. A. Urpi, M. A. Bonuccelli, and S. Giordano, "Modeling Cooperation in Mobile Ad Hoc Networks: A Formal Description of Selfishness," in *Proceedings of WiOpt 2003,* pp. 303–312.
32. J. Vaucher, P. Kropf, G. Babin, and T. Jouve, "Experimenting with Gnutella Communities," in *Distributed Communities on the Web (DCW 2002),* Sydney, Australia, April 2002.
33. D. J. Watts, *Small Worlds: The Dynamics of Networks between Order and Randomness,* Princeton University Press, Princeton, NJ, 1999.
34. S. Zhong, Y. R. Yang, and J. Chen, "Sprite: A Simple, Cheat-Proof, Credit-Based System for Mobile Ad-Hoc Networks," Technical Report, Yale/DCS/TR1235, Department of Computer Science, Yale University, July 2002.

CHAPTER 14

SIMULATION AND MODELING OF WIRELESS, MOBILE, AND AD HOC NETWORKS

AZZEDINE BOUKERCHE and LUCIANO BONONI

14.1 INTRODUCTION

Mobile ad hoc networking technologies and wireless communication systems are growing at an ever faster rate, and this is likely to continue in the foreseeable future. Higher reliability, better coverage and services, higher capacity, mobility management, and wireless multimedia are all parts of the potpourri. The evolution of new systems and improved designs will always depend on the ability to predict mobile, wireless, and ad hoc networks' performance using analytical or simulation methods. Modeling and simulation are traditional methods used to evaluate wireless network designs. To date, mathematical modeling and analysis have brought some insights into the design of such systems. However, analytical methods are often not general or detailed enough for evaluation and comparison of various proposed wireless and mobile systems and their services. Thus, simulation can significantly help system engineers to obtain crucial performance characteristics.

However, detailed simulations of these systems may require excessive amounts of CPU time, and their execution on sequential machines has long been known to have computational requirements that far exceed the computing capabilities of the fastest available machines. For instance, it is not unusual for simulations of large wired and wireless networks to require hundreds hours or even days of machine time. As a consequence, the development of methods to speed up simulations has recently received a great deal of interest [6, 7, 8, 9, 22, 38, 65].

With the ever increasing use of simulation for designing large and complex systems, wireless mobile and ad hoc networks have brought several challenges to the parallel and

Mobile Ad Hoc Networking. Edited by Basagni, Conti, Giordano, and Stojmenovic.
ISBN 0-471-37313-3 © 2004 Institute of Electrical and Electronics Engineers, Inc.

distributed discrete-event simulation (PDES) community. The challenges require not only extension of and advances in current parallel and distributed simulation methodologies, but also the discovery of innovative approaches and techniques to deal with the rapidly expanding expectations of wireless network designers [6].

In this chapter, we shall present some guidelines related to Mobile Ad Hoc Networks (MANETs) modeling and simulation, several sequential network simulation testbeds, and distributed simulation testbeds for wireless and mobile networks. We shall also address the challenges PDES community has to face in order to design high-performance simulators.

14.2 DESIGN AND MODELING OF WIRELESS AND MOBILE AD HOC NETWORKS

In this section, we shall introduce the basic characteristics and major issues pertaining to MANETs' modeling and simulation. A complete definition of all aspects of interest and a fit-all solution for simulation is not possible and out of the scope in this chapter, because it depends on the objectives sought by the designers and the assumptions they have made. Thus, we will point out only some of the most promising and interesting challenges and solutions to the modeling of mobile ad hoc networking and communications, and we will present the state of the art of simulation of wireless, mobile, and ad hoc systems. The main purpose of a simulation-based study for a MANET system is to obtain detailed information about performance figures, behavior, overheads, quality of service, and many other metrics regarding the system, protocols, and policies adopted at many levels of the ISO/OSI protocol stack [12, 18, 23, 29, 56]. Among all the model parameters, one defined as "factor" is selected as the varying parameter whose effect on the performance indices is to be evaluated [29]. Evaluating system performance via modeling and simulation consists of two preliminary steps: (1) defining the system model, and (2) adopting the appropriate simulation technique to estimate the metrics needed to evaluate the performamce of the system. In what follows, we will first talk about MANET modeling. Some concepts can be considered general for every wireless and mobile system (e.g., wireless PCS, cellular networks), whereas others can be considered specific to MANETs.

14.2.1 Mobile Ad Hoc Network Modeling

As stated before, it is not convenient to talk about MANET models without defining the set of objectives and questions the simulation experiments should answer to. Every system model is tailored depending on the goal of the simulation project. Any unrequired additional detail will introduce overheads, possible errors and a slowdown of the simulation process [26]. Any missing detail relevant for the performance evaluation of the system will also introduce errors and lead to approximated results and the need for additional model updates [15, 26]. General-purpose models are known to be very complex and hard to adapt to specific system models. Today, many simulation tools provide a library of simulation models written by professional modelers and researchers [20, 45, 46, 47, 55]. Many times, when incremental updates are performed by different people, the model validation becomes a time-consuming and difficult task to overcome [15]. Most of the times, models are supplied or exchanged without any comments and/or documentation, requiring a great effort of designers to interpret and validate them [15]. Today, most of the models can be defined by using *object-oriented paradigms* and *languages,* such as Java, C/C++, and OTcl/Tk, just to mention a few. This makes it possible and more practical to extend, adopt, exchange, and reuse existing models in new simulation projects. Inheritance allows us to create module hierarchies and instances of complex objects, with a simple management of model libraries. Widespread adoption of object oriented-languages works in favor of model distribution among researchers. C++ models and tools are usually adopted and preferred to Java-based tools for simulation-performance purposes. Modeling component-based units can be performed with the adoption of high-level compositional languages and a set of application tools [45, 46, 47, 55].

In the performance evaluation of a wireless, mobile ad hoc system, every simulation experiment should be done under a variety of modeling conditions and factors, in order to capture detailed and "realistic" effects of the real system. These conditions should be well defined and may have wide and clear interactions at various levels in the model. As an example, correct model design should evaluate a priori any possible relationship among physical, topology, and mobility levels, up to the protocol layers such as Medium Access Control, Logical Link Control, Network, Transport, and Application. Examples of such conditions include transmission ranges, power consumption, detection thresholds, data-traffic sources and loads, buffering storage, user mobility and topology restrictions, signal propagation and obstacles, interference and bit errors, just to mention a few.

Figure 14.1 shows a roadmap of some modeling issues that will be considered in this chapter. The edge-based representation of the multiple effects, relationships, and conditions among the modeling issues is shown just to put in some evidence the nontrivial work of the modeler. Also, Figure 14.1 emphasizes that mobility plays a central role in the model design. All of these conditions need to be represented and managed within the system model, by means of well-defined and efficiently manageable data structures. Many solutions which model the test conditions, in many simulation studies, have been proposed in the literature for wireless and mobile system models, and will be discussed in this chapter. Specifically, simulation models for wireless and mobile systems in general, and MANETs

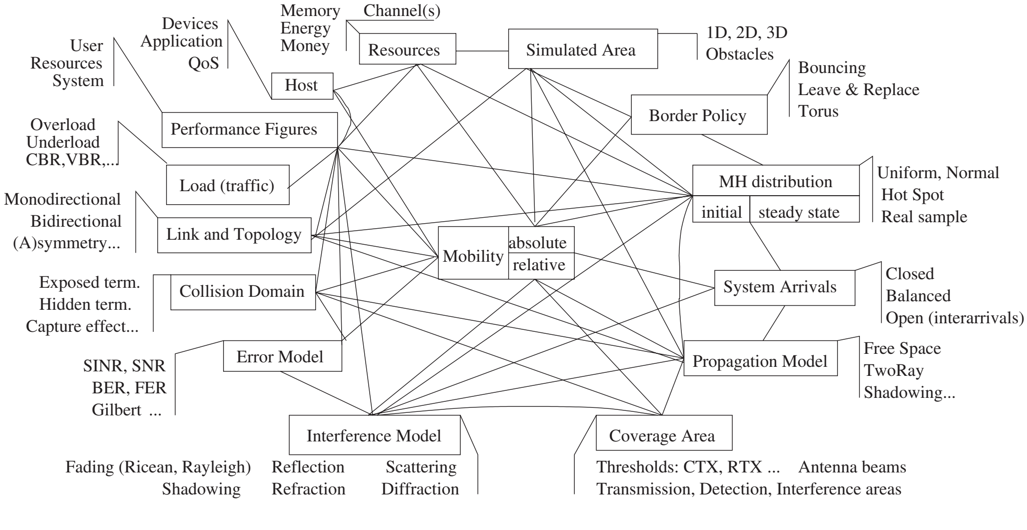


Figure 14.1. The modeling roadmap.

in particular, have to deal with at least two innovative concepts with respect to wired network models: the *user-mobility* and *open-broadcast nature of the wireless medium*. In this chapter we will discuss the model definition issues related to these innovative concepts for wireless mobile systems, with a special attention to MANETs. Our presentation starts from the bottom and proceeds to the upper levels, that is, from physical, topological, and mobility models, up to protocol layers. The model implementation would depend on the model-definition languages and simulation techniques and tools adopted, thereby requiring additional validation and verification efforts. Hence, we will not discuss in detail the model implementation, verification, and validation in this chapter.

14.2.1.1 Simulated Area and Boundary Policy. In this section, we shall discuss the simulated area and the boundary policy concepts, as well as a relevant set of assumptions related to the design and modeling of the simulated area, which may have many effects on the simulation of the target system [3, 12, 13, 32, 55]. This area is the virtual theater of execution of the simulation, and the area size is not important in this discussion. The area size becomes important when coupled with other parameters governing the mobility, propagation, and node-density models considered in the proposed scenario.

First of all, the *simulated area* of interest is limited (i.e., it is bounded by limit borders); it can be mono- or multidimensional, and can be represented by Cartesian coordinates[1] as follows:

- *Monodimensional (1-D)* area is a simple linear path for a set of MHs (e.g., a simple highway-simulation model). The relevant parameter is one single x coordinate along the linear path, varying in the limited range [min$X$, max$X$]. Such a model can be used in cellular systems, assuming a linear path between a set of adjacent cells, and it is infrequently used in MANET system simulation.
- *Bidimensional (2-D)* areas are the most used models because they allow us to embed and map any possible user path in a real (flat) geographical area. Every portion of a real geographical area can be mapped on a 2-D grid with (x, y) coordinates varying in the limiting ranges [min$X$, max$X$] and [min$Y$, max$Y$]. Definition of subgridding cells can be exploited to manage and sample object distributions. As an example, hierarchies of grid cells can simplify the management of "neighbor" objects in adjacent cells, and can support object distribution policies (e.g., n objects per grid cell). When object density evolution is required to be evaluated, grids allow a consistent, snapshot-based sampling and runtime calculation of the objects' distribution.
- *3-D models*. Sometimes, 2-D models can be extended to three-dimensional space models (x, y, z); for example, when modeling user mobility inside buildings with many floors, user mobility can be described by including vertical movements, as in staircases and elevators [34].

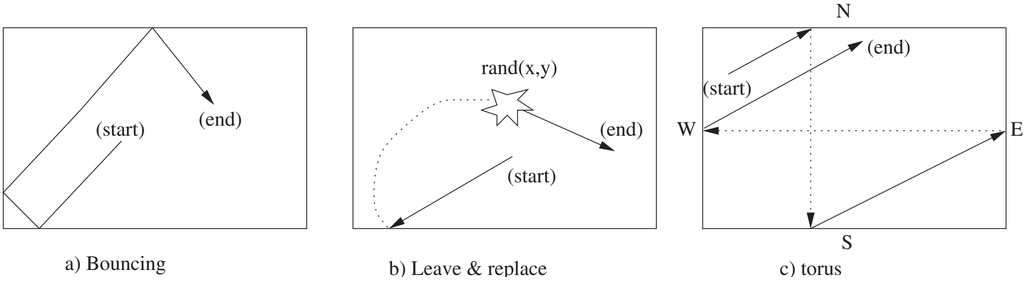
The simulated area may also be enriched with obstacles, affecting user mobility and propagation of signals. A brief discussion will be presented in the following sections about propagation and mobility models. Obstacles can be modeled and realized as additional data structures.

The *boundary policy* is another relevant characteristic of the simulated area that one might consider in the model design. This policy defines the behavior of the mobile hosts

[1] Note that polar coordinates can also be used.

(MHs) when, due to the motion process, they reach the boundaries of the simulated area. Many possible solutions have been proposed depending on the simulation and model requirements. The modeler should be careful about the effect that the boundary behavior, composed with a mobility model, may have on the resulting spatial node distribution [3, 4, 5, 13]. In what follows, we shall distinguish between three boundary policies widely used by the wireless and MANET networks' modelers (Figure 14.2):

- The *"bouncing boundary"* solution requires mobile hosts (MHs) to bounce back toward the simulated area when they are going to move outside [3, 5, 13] (see Figure 14.2a). This simple solution can be used if the number of MHs in the simulation is required to be constant (e.g., a closed system). Different "bouncing" rules can be adopted, for example, *mirror reflection* (e.g., preserving the angle of incidence θ in a bouncing angle $-\theta$ or $\pi - \theta$, respectively) and *random reflection* (i.e., a random reflection angle θ is generated). This boundary policy can be considered as quite unrealistic for large areas, and quite approximated for simulation of indoor mobility (e.g., people moving in a room). A modeler has the choice to either preserve the state of a bouncing MH or to create a new instance of MH when it virtually "leaves" the area (i.e., the MH hits the area boundaries). This choice may be useful if the state information of the MH is relevant for the simulation process or for the protocol to be tested (e.g., protocols based on MH's history and evolution state).
- The *"leave and replace"* variant of the bouncing boundary policy is to delete a "bouncing" (leaving) MH and clone it in a randomly chosen position within the simulated area, following any node-position distribution (see Figure 14.2b). As in the "bouncing boundary" solution, the cloning of the leaving MH can preserve state and history information on the new instance or not, depending on modeler choice. This policy may result in a nonuniform steady-state spatial node distribution, with a node concentration around the center of the simulated area [3, 5, 13]. Intuitively, this is due to the fact that MHs leave from the boundaries and reenter by choosing a randomly distributed position "inside" the area. This also results in a biased (i.e., reduced) probability of finding MHs moving from the borders toward the middle of the simulated area. This solution is rarely considered useful in MANET simulation.
- The *"torus boundary"* solution is another policy widely adopted by many researchers [3, 13]. In this policy, when a mobile host reaches the north, west, south, and east boundaries of any rectangular area, it simply leaves the area and reenters with the same direction and speed from the south, east, north, and west bound-



a) Bouncing b) Leave & replace c) torus

Figure 14.2. Three boundary policies.

aries, respectively (see Figure 14.2c). Intuitively, the rectangular area is wrapped around itself, north with south, and east with west, like a ring. The reason why the torus policy is widely used is given by its simple implementation, and because it simplifies the management of uniformly distributed host densities and directions (accordingly with the implemented motion models). There is a full correlation effect balancing leaving and reentering hosts' directions and velocities. Again, leaving MHs can preserve state and history information when they reenter the area, depending on modeler choice (except for velocity and direction, which should be maintained).

14.2.1.2 Host Sources and Position Distributions. Host mobility is the main factor in determining the "arrival" and physical presence of a set of MHs within a fixed simulated area. The physical presence of hosts does not necessarily represent the relevant factor to be modeled for the simulation goals. This mainly depends on the MH roles to be modeled and on the performance indices required. As an example, a switched-off MH in the simulated area may not be considered relevant to determine transmission-based performance indices. We denote as "active" the role of a MH that can be considered effective in determining the value of a performance index whose evaluation is the goal of a simulation run. Every performance analysis for a wireless mobile system can be performed by assuming a (fixed or variable) number of "active" mobile hosts (MHs) implemented in a limited area of interest. In what follows, we focus upon active MHs, and consider only the motion-related "presence" of these MHs. The modeler should evaluate the opportune and realistic definition of the average density of "active" MHs, with attention to the policies for the creation and position allocation of new MH instances, both at the simulation start and at run time [54].

Dealing with the sources and creation policies in accordance with the factors influencing the performance metrics of interest, one possible choice is to require the number of active objects in the simulated area to be constant. This choice may be useful, for example, when performance metrics to be obtained are related to the number of MHs, or to the MHs' density (e.g., existence of route path, network partition, average degree of MHs). In the following, we denote MHs' presence in the simulated area as "active presence," whatever meaning this would imply:

- In *closed systems,* the initial number of "active" simulated MHs is constant, and every MH lives (i.e., it maintains its functionalities) for the whole simulation run.
- *Balanced systems* realize a simple hybrid solution that can be useful for some simulation analysis. Every time a MH leaves the simulated world (e.g., moving outside the simulation area, or switching off the network interface), a new instance of the MH is causally introduced in the simulated area, following the selected position distribution. Bouncing, torus, and "leave and replace" boundaries allow a natural implementation of a closed or balanced system under the MHs' mobility viewpoint (if MHs' sources and sinks are missing).
- In *open systems,* one or more sources of active simulated MHs are defined (e.g., interarrival or activation processes for MHs). In such scenarios, the modeler should evaluate and select interarrival time distributions for the sources (e.g., exponential or Poisson distribution), sink policies (e.g., MHs with no battery energy or moving outside the simulated area are discarded), and the initial-position distribution for in-

coming MHs (e.g., at random uniformly distributed coordinates in the area). If the arrival process is too fast, the system is *unstable,* that is, the asymptotical number of MHs in the simulated area is not upper bounded. This can lead to a biasing problem if we are interested in the evaluation of performance metrics that may be related to the average MH density (e.g., multihop link reliability in routing protocols, network connectivity and partitions, average next-hop distance, average transmission power, etc.) [3, 12, 13, 23]. To obtain an open, stable system, the rate of MHs leaving the area should be statistically balanced by the sum of arrival rates of the MH sources. This means that the number of MHs in the area is not a constant value (as in closed and balanced systems), but converges asymptotically.

One possible choice for *initial allocation* of a new MH's position is a random selection of its position coordinates within the simulated area. Uniform or normal distributions are widely used in the literature, depending on the host density to be modeled (e.g., uniform vs. hot-spot density, respectively) [13, 55]. This choice does not provide any best-effort guarantee about network partitions. One possible solution to this problem is to divide the simulation area into a grid of square cells, with a size that could be determined by the minimum range of connectivity among the MHs, and distribute MHs' positions such that at least one MH is in every cell of the grid.

On the other hand, a real, sampled distribution snapshot of MH positions can be used, if available. This is a common way to model hotspot MHs' distribution [55]. In many scenarios, depending on the mobility models and their respective parameters adopted for MHs, the initial distribution of MHs is less or more relevant in determining the steady-state MHs' distribution [3, 13]. When the "memory effect" of the initial distribution is not preserved by the motion model characteristics, every possible transient effect of the initial distribution should be evaluated and eliminated to collect unbiased steady-state results.

The choice of *runtime position-allocation* policies for newly generated MHs is a little bit more subtle. Random allocation of MHs may result in nonuniform distribution and biased node density, for example, if hosts leave the system only from the boundaries [3]. A possible solution for this scenario would be to "delete" hosts selected randomly in the simulation area, and to adopt distribution-balancing boundary policies such as bouncing and torus borders [3, 5, 13].

14.2.1.3 Coverage Areas, Physical Propagation, Transmission Errors, and Interference Models. Usually in MANETs, every host can be considered as a potentially mobile host. As a consequence, hybrid MANETs under analysis today may include fixed, static base stations (BSs), with their respectively managed coverage areas, as in cellular and PCS systems [18]. A detailed physical model for wireless transmission, including propagation, mobility, error, and interference models, is one of the most difficult and computationally expensive tasks to do, and strong approximation and assumptions are usually introduced [2, 3, 12, 35, 42, 62, 66]. Many models and solutions have been proposed, at different levels of detail [26, 59]. We will skip most of the details, for space reasons, and we will just point out some of the modeling issues related to MANETs.

The physical wireless transmission is based on the emission of electromagnetic waves coding information with many possible modulation and coding techniques. The natural decay of transmitted signals can be modeled following simple analytical approximations. If the residual signal power of the receiving network interface is above the *detection* *threshold,* a communication is possible. Otherwise, to allow a communication (link establishment) between the intended sender and receiver, it would be necessary to increase the transmission power of the sender and/or reduce their relative distance *d*.

One of the most used propagation models, adopted in MANET simulation is the simple *Free Space Propagation Model* [12, 53]: if *Pt* is the transmission power (i.e., energy/time) used for the signal transmission, then the receiving power *Pr* is proportional to $1/d^2$, where *d* is the distance between sender and receiver in open space (see Figure 14.3).

The Free Space Propagation Model can be extended to better describe the effects over near and far receivers, with the *Two-Ray Ground Reflection Model* [12, 53]. This model is the same as Free Space Propagation Model, except when the distance *d* is greater than a crossover point, called the *reference distance* (around 100 meters). For such long distances, the receiving power *Pr* is modeled as proportional to $1/d^\beta$, $\beta > 2$.

The Free Space and Two-Ray propagation models assume ideal propagations over a circular area around the transmitter. To model irregular coverage areas, the *Shadowing Propagation Model* [45, 53] is defined with two components: a component similar to the Free Space Propagation Model, and a random component to make randomly variable (and statistically controlled) the edge of the communication range. For a complete discussion of the Free Space Propagation Model and other models, see [26, 53].

A modeling choice to define if a transmission can be detected by a tagged receiver is to define a *receiving threshold* (*RTX*) and a *carrier-sense threshold* (*CTX*) for every device [12, 59, 53]. For every simulated transmission, it would be required to scan every MH in the system and to apply the propagation model to the transmitted signal. This requires evaluation, for each receiver, if the receiving power perceived for the ongoing transmission is sufficient for reception (i.e., greater than RTX), if it is sufficient for detection and carrier sensing (i.e., greater than CTX), or if it is simple interference. Reception and carrier sensing events can be passed to the model components devoted to manage events at the upper layers of the model, for example, Medium Access Control policy implementation. This scan-based computation may require a long time if performed for a large set of MHs.

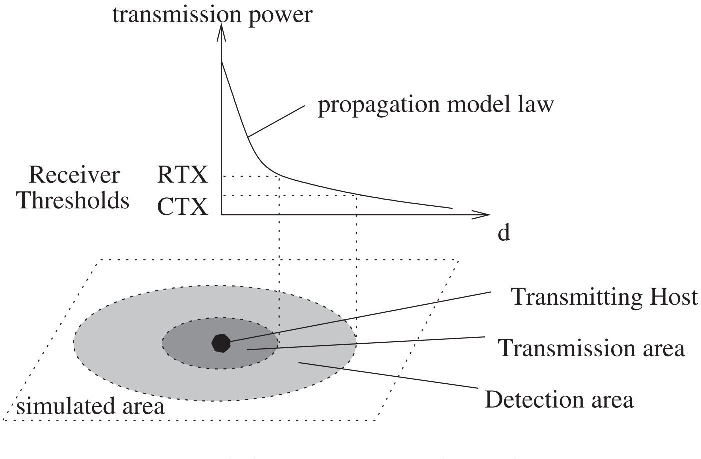


Figure 14.3. Transmission power, propagation, and coverage areas.

The system model can be extended with *coverage areas,* in order to reduce the transmission-detection overhead and to model much more complex propagation models, depending on the modeling and simulation requirements. Transmission coverage area definition can be directly associated with every transmitter, but the area size and shape is relative to the receiver thresholds. For ease of management, the area-size definition would require the assumption of common threshold levels (i.e., common CTX and RTX values) for every MH in the system (see Figure 14.3).

The *transmission (coverage) area* of a wireless transmitter can be defined as the area where the transmitted wireless signal propagates and can be correctly detected and decoded (i.e., transmission is possible with few/no errors due to interference). This area depends on the transmission power of the transmitter, on the propagation model, on the reception threshold (sensitivity) of the receiving network interface (RTX), and on the amount of interference (noise) caused by many possible factors (described in the following). The transmission area should be defined and managed in the model, for each MH, in order to dynamically evaluate a communication capability (i.e., a direct link) between every candidate transmitter and receiver.

The *detection area* (see Figure 14.3) of a wireless transmission device is the area where the signal propagates, and where it can be detected by a carrier sensing mechanism, without being necessarily decoded (i.e., $CTX \leq Received\_Signal\_Power \leq RTX$). This means that a mobile host can sense the wireless medium as busy without being able to decode received signals. The definition of this area in the model, for each MH, may be relevant for the evaluation of detailed carrier sensing and MAC-level effects, such as exposed terminals, hidden terminals, and capture effects (described in the following) [24, 52].

The *interference area* of a wireless transmission device is defined as the area where the transmitted wireless signal propagates, without being detected or decoded by any receiver, adding interference and noise to any possible ongoing transmission for intended receivers. The cumulative effect of noise (i.e., interference) might add errors to the transmission of bits of information. The definition and management of this area in the model, for each MH, may be relevant for the evaluation of detailed interference and error models. Transmission areas are included in detection areas, and detection areas are included in interference areas, given the propagation properties of wireless signals in open spaces. Possible choices to model coverage and interference areas in open spaces are given by regular polygons centered in the transmitter's position. Circular coverage areas can be defined for open-space propagation models, and are a simple common choice for MANET modeling and simulation purposes. The circle radius, centered on the transmitter position, can be made proportional to the transmission power in an adaptive way, mapping to real power control management and policies implemented in simulated MHs. If fixed (static) transmitters are present, with a constant transmission power, as in cellular and PCS networks, a common choice is to approximate the coverage areas by hexagons, squares, and Voronoi diagrams. This choice can simplify the link management, because connectivity between a MH and a fixed transmitter can be evaluated with no ambiguity (i.e., only one reference transmitter is defined in every point). This choice is simple to define and to manage for the simulation purpose, but it realizes a strong approximation of the real behavior of wireless transmissions. The circular coverage model is quite realistic in open spaces, but it would require some changes if obstacles were present to interfere with signal propagation.

When *obstacles* are present in the simulation area, the coverage and interference areas may be severely affected, in almost unpredictable ways [35, 47, 53]. This is even worse if mobility of wireless sources (or, equivalently, wireless receivers) is present. Models to deal with obstacles have been defined [32, 47, 53, 55]. Moreover, if the antennas are not omnidirectional and the transmission beam is not *isotropic,* the regular polygon choice for coverage areas is even more approximate (e.g., with smart antennas, the transmission energy is not uniformly propagated in all directions, but has a directional effect [53]). Modeling of asymmetric beams and obstacles in these scenarios might be quite complex and computationally expensive, too.

To model *interference effects,* many additional factors need to be defined, and a realistic model is quite hard if not impossible to obtain, without paying for a high computation overhead. As a simple description of the wide set of problems encountered when dealing with accurate interference modeling of physical wireless transmission, we present a short list of system details that should be considered in theory, and that aren't usually considered in many models, given the complexity and extensive computation required to simulate their effects. Most of the described problems are given by continuous physical phenomena, whose approximate modeling in the discrete-event simulation field would be really hard and expensive. Mobility is an additional source of problems. In MANET scenarios, the model would be even more complex than in cellular and PCS networks, because both the transmitter and the receiver usually can be mobile, and a distributed, *relative-mobility* parameter should be evaluated, instead of a local, absolute-mobility parameter; Physical problems to be modeled in wireless transmission include the following phenomena [53]:

- *Fading:* a physical phenomenon, frequency dependent, inducing delay and phase variations between the main transmitted signal (following a dominant path) and many secondary signals (following alternative paths), caused by obstacles and mobility. This causes long-term and short-term variations of the resulting reception power of transmitted signals. The *Additive White Gaussian Noise* model is used to represent ideal channel conditions under the signal fading viewpoint. *Rayleigh* and *Ricean* Fading are widely accepted models [2, 35, 59, 53] used for fading-prone scenarios. They can be applied to highly mobile scenarios, No Line of Sight (NLOS) and Line of Sight (LOS) paths, respectively [59]. The *K* parameter of the Ricean Fading Model can be used to control the composed effect of LOS and NLOS signal powers [59]. A *Coherence time* parameter is adopted to control the time frequency and duration of fading effects on the channel. As an example of the modeling complexity for fading effects, let us assume that there are M base stations and N mobile hosts in the simulation scenario, and there are roughly L paths determined by obstacles in each propagation direction. Then we would need approximatively up to $2M * N * L$ instances of Rayleigh fading generators. Efficient implementation of fading models is still an ongoing research activity [2, 35, 53].
- *Shadowing:* attenuation of signal power propagation caused by physical obstacles. This effect is mainly responsible for irregular coverage areas. The Shadowing Propagation Model defined in the previous section gives a statistical approximation of this effect [53].
- *Reflection:* signal reflection caused by large obstacles and indoor walls. It is quite important to model this effect for indoor scenarios.
- *Refraction:* Marginal signal change and reflection caused by variation in the medium density.
- *Scattering:* signal diffusion caused by sharp obstacles.
- *Diffraction:* signal deviation caused by large edges and corners.

Each one of the above-mentioned phenomena may have different characteristics, given different physical implementations and different coding techniques adopted for wireless transmissions. The whole effect of such a collection of complex phenomena is usually modeled as a simple error probability for a given amount of information received (e.g., a bit) on the physical channel. The idea is to enclose all this in a black box describing the whole effect and call it the *probability to obtain a bit error*. Obviously, this may be a strong, unacceptable approximation, depending on the aim of the simulation. In many models, in order to approximate the real behavior of the wireless medium, the physical medium (or its high-level abstraction, the channel) behavior can be described with more accurate error models. Signal to interference and noise ratio (SINR) and signal to noise ratio (SNR) are the key parameters adopted to model the signal composition of interference effects described above. The generalized SINR and SNR values, together with RTX and CTX thresholds, can be adopted to model with some detail the high-level effect of interference resulting in bit error rates (BER) and frame error rates (FER). BER and FER values model the generalized probability that a transmitted bit or frame is received with errors, respectively. FER is a function of BER and the frame length (in bits). In order to capture the bursty effect of wireless transmission errors, the *Gilbert–Elliott Error Model* has been used to define the status of the wireless medium as a function of time [62]. This model defines a random Markov process between the following two states: *good* and *bad*. Good status is characterized by low probability of bit error (low bit error rate, BER), whereas bad status is characterized by high bit error rate. The time in bad and good status is usually sampled from an exponential (or its discrete counterpart, geometric) time distribution, with respective parameter and average values. It is a good choice to implement it in time-slotted models.

The coding techniques adopted for the wireless transmission and the frequency spectrum allocated for adjacent channels are additional parameters to be evaluated in order to define opportune error models for wireless communication. New coding techniques allow for interference effects' cancellation, and interference models should be defined for adjacent-channel interference and co-channel interference [53]. We skip all the details for space reasons.

14.2.1.4 Link Definition and Network Topology

The modeling of coverage areas is really important because it is related to the link definition between any couple of MHs, or between a MH and a BS. This is really important in MANETs, because a simple star topology given by a set of MHs around a BS is not a concrete and dynamic vision of the system. Moreover, in modeling and simulation of MANETs, the "link-established" property between a couple of MHs is more complicated than in most wireless cellular networks, mainly due to the management of relative mobility (as opposite to absolute mobility) of these mobile hosts [66]. For any couple of MHs and/or fixed hosts (e.g., base stations) covering the area of interest for the simulation (call them host X and Y), we have three possible expected scenarios related to reception and carrier sensing thresholds, coverage areas, and link definitions [24]:

1. X is out of the transmission area of Y, and vice versa. This means that X and Y are partitioned (e.g., see A and C in Figure 14.4). The network topology is not assumed to have any direct communication link between X and Y. Maybe communication between X and Y is possible, at the upper routing layer, if supported by an intermediate-hosts chain of mutually reachable hosts (e.g., like host B and C in Figure 14.4).

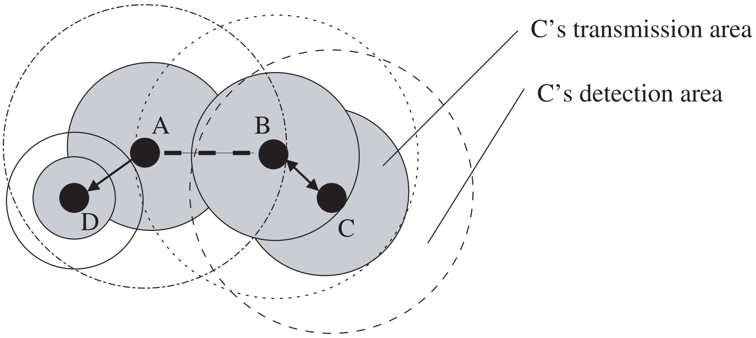


Figure 14.4. Example of a collision domain.

2. X is within the transmission area of Y, and Y is out of the transmission area of X. This means that Y can communicate with X, but not vice versa. In this scenario, a monodirectional link exists from Y to X (e.g., see hosts A and D in Figure 14.4). Monodirectional links exist in many real scenarios, mainly due to the different transmission power and propagation characteristics of MHs (e.g., see host D in Figure 14.4). The obtained network topology is a direct graph based upon the monodirectional links.
3. X is within the transmission area of Y, and vice versa. X and Y are mutually reachable via a wireless bidirectional link (e.g., see hosts B and C in Figure 14.4). Depending on the coding techniques and channel bandwidth allocated for the physical channel, it may happen that the bidirectional link is not a symmetric link. A bidirectional link is *symmetric* if the physical channel capacity (i.e., the maximum bit rate obtained for wireless transmission) is the same for both link directions (otherwise it is asymmetric). Many simulation models usually assume bidirectional and symmetric links, for ease of implementation. The assumptions about these scenarios may severely influence the modeling and performance results in the evaluation of network protocols, for example, in routing protocols and multihop communication.

Dealing with discrete, event-based simulation, the critical question related to MANET topology management in the simulation process is *what is the simulated time of next link-state-change event that will be expected, given the current relative mobility pattern and coverage areas of mobile hosts?* The answer to this question would require a little more computation, based on the model and data structures defined to implement the simulation. Every link-state-change event can be calculated, based on current coverage and mobility conditions, and its execution scheduled in an ordered event list. Any intermediate change in the speed, direction, or transmission power of any one of the involved hosts would require us to delete the causally dependent, scheduled link-state events and substitute them with the updated ones. Moreover, this event-list management is at the basis of any discrete event simulation. This indicates clearly how complex and computationally hard the mobility and link management in MANETs can be.

Once the link existence is established, many other conditions of the high-level link properties should be managed. As an example, in wireless physical channels it is not possible to receive a communication on the channel while a simultaneous transmission is performed on the same physical channel [24]. A bidirectional (full duplex) link can be obtained by adopting time-division duplex (TDD) or frequency-division duplex (FDD). TDD consists in splitting the transmission and reception phases over adjacent, non-overlapped time intervals on the same physical channel. FDD consists in adopting two physical channels: one for transmission and one for reception. In MANET modeling and simulation, time-division duplex is commonly adopted. All data transmissions and receptions have to be in the same frequency band, since there are no "bridge" nodes (perhaps except base stations) to translate the transmissions from one physical channel to another one. This usually requires strict time synchronization in the system, and Medium Access Control (MAC) protocols definition [24]. Frequency-division duplex may be adopted (together with TDD) in centralized networks (like cellular) characterized by up-link and downlink channels [24].

The coverage area management in the model can be used to simulate additional MAC level details relevant for MANETs, such as *collision domains*. A collision domain can be defined as the coverage area shared by a set of MHs mutually connected by a single shared communication channel (i.e., a single logical channel). Collision of concurrent signals transmitted on the same collision domain would cause a destructive interference for detected signals on the receiver. The main task of a MAC protocol policy is to avoid such collisions, mainly by avoiding the start of new transmissions while another transmission is being detected. A detection-based policy for a MAC implementation may result in some problems whose investigation would require an accurate coverage-model definition. As an example, let us suppose A is within the *detection area* of B and vice versa, B is within the *transmission area* of C and vice versa, and C is outside the *detection area* of A (see Figure 14.4). In this scenario, A senses the transmission of B, that is, it senses the channel as busy, but it cannot decode the transmission. This condition is often modeled in order to obtain a real performance investigation about *exposed terminals* [24]. Exposed terminals are terminals, (e.g., A) whose transmission is not allowed (e.g., by a MAC policy over a collision domain) due to exposure to irrelevant transmissions, (e.g., B to C). A similar problem is given by *hidden terminals:* due to shadowing effects and limited transmission ranges, a given terminal C could start a transmission toward another terminal B (C and B are within each other's transmission area; see Figure 14.4) while B is receiving signals from a hidden (with respect to C) terminal A. This means that B cannot complete any reception, due to the destructive collision of signals from A and C. It may also happen that B can detect and isolate one of the colliding transmissions, (e.g., from C to B). In this case, we model a *capture effect* of transmission from C to B, despite A's interference and collision. A discussion of details for hidden and exposed terminals and modeling of capture effects can be found in [24, 52]. A rough modeling, based on a shared, global, Boolean variable, *Channel* = *Busy* or *Idle,* would not describe with required accuracy the previous scenarios.

This discussion was given to illustrate how, in the simulation plan, the definition and the structure of the coverage area, topology, and interference models may include the information required to perform a realistic simulation. Anyway, detailed models are quite complex, and simplifying techniques and assumptions are widely adopted. In the following, we are going to describe another relevant characteristic of MANET and wireless network models adding additional complexity to the model definition and management: host mobility.

14.2.1.5 Mobility Models. User mobility is the main added value of wireless networks. Accurate simulation results would require accurate details to be modeled, and many fine-grained, low-level causal effects to be taken into account in the simulation process. Mobility has a central role, and is a relevant background effect to be modeled in almost every simulation analysis of wireless systems (see Figure 14.1). The effect of mobility on the system policies and protocols is relevant at many layers: dynamic topologies, due to simulated hosts' mobility, map causality effects in the "areas of influence" of each mobile device, resulting in dynamically shaped causality domains [6, 24]. The effect of mobility introduces adaptive behaviors of users, protocols, and applications. Moreover, it may happen that mobility models are related to the physical scenario under consideration [55, 60]. Mobility patterns may sometimes be application dependent [13, 55]. Most of Medium Access Control, Routing, and Transport protocols proposed for MANET scenarios are customized and designed for specified mobility models, and behave better than a general-purpose protocol for that given scenario. The evaluation of user positions can be a computationally relevant task in a wireless mobile system's simulation, due to the mobility and high number of events related to the user position. In two or more neighbor hosts simply sharing the wireless medium (without any end-to-end communication session on), the causal effect of signal interference, due to mobility, could result in a chain of local-state events from Medium Access Control (MAC) up to the Transport and Application layers [23, 59]. In MANETs, given the infrastructureless architecture, some of the mobility models adopted for cellular systems are not appealing. As an example, *Markov models* (random walks) described by cell-to-cell migration probabilities, or *fluid-flow models,* whose characteristic is to describe host mobility in terms of "the mean number of users crossing the boundary of a given area," are not considered as relevant as random- and restricted-mobility models, gravity models, or group-mobility models.

In general, two types of mobility models can be adopted in the simulation of wireless mobile networks, and specifically MANETs: *motion traces* and *synthetic models* [13].

Motion traces provide accurate and realistic information about user mobility patterns and behavior, in particular when user mobility is related to real users in a bounded scenario (e.g., downtown streets, highways) [3, 13, 55, 60]. Unfortunately, traces require large log files, depending on the number of tracked hosts and the time granularity of samples. Traces are significant descriptions about the steady-state mobility of a user only if the motion samples are collected for significant time intervals. If the sample frequency is low, approximated solutions (e.g., interpolation and dead reckoning) can be used, but this requires additional computation, and may result in strange behavior (like users walking through obstacles instead of turning around them). Moreover, traces can be collected only for existing systems, and MANET traces are still hard to find mainly because large MANETs scenarios have still to be implemented and user applications have to be defined. Motion traces solve the problem of defining the initial and run-time position distribution of MHs in a deterministic way. Another interesting characteristic of motion traces is their ability to capture the real correlation effect between user mobility and real application/user needs. It may happen that user movement is driven by application needs, for example, in order to reach good coverage areas. Also, it may happen that users move by showing a correlated group behavior [13, 28, 61]. Synthetic models trying to define a similar correlation effect, for single MHs and MH groups, will be defined in the following.

Synthetic models are defined to represent the mobility of users in a realistic way, without using traces. Many synthetic models have been defined in the past to be adopted as analytical models [3, 13]. The main requirement for such analytical models was mathematical tractability instead of realism. Such models also survived in many simulation studies, mainly due to validation possibilities they offer with respect to the simulation counterpart [5]. In other scenarios, random-motion models far from reality can be adopted in order to stress a given mechanism or protocol, emphasizing worst-case scenario results [13, 27]. Random-motion models have been recently extended by introducing correlation effects, restrictions, and group behaviors, in order to meet the requirements of mobile systems' modeling and simulation [3, 13]. We can distinguish between three degrees of "randomness" in the classification of random models [3, 5]: (1) models that allow users to move anywhere in the simulation area, following pseudorandom processes to select speed and direction, (2) models that bound the movement of users (like streets, walls, etc.) but still allow for pseudorandom selection of direction and speed at crossings (like City Section and Manhattan Model [42]), and (3) models based on predefined paths (deterministic paths).

In the following, we define and discuss a list of synthetic models used for MANET simulation.

- *Random Mobility Model.* This is a discrete interpretation of the Brownian motion model [3, 13]. It is completely unpredictable and it has the memoryless property for speed and direction. This means that current speed and direction are not related to the speed and direction history, and speed and direction are completely uncorrelated (i.e., two independent, stochastic processes) [3]. Many mobile users adopting such motion model result in completely uncorrelated mobility, and the mobility pattern is quite unrealistic. This is a typical worst-case modeling assumption, for example, when the analysis demonstrates that the adaptive protocols' performance does not rely on any motion correlation and/or predictable-position assumptions. This model can be used for vehicular and large-scale environments, and can be implemented in many ways. Assuming a 2-D motion, one possible implementation is the following: every MH moves from a current location to its new location defined by a uniformly distributed pseudorandom choice of a new direction θ in the polar angle interval $[0, 2\pi[$, and by a uniformly distributed pseudorandom choice of a speed value s in [*minSpeed, maxSpeed*]. The speed and direction are maintained for constant time values *ts* and *td,* respectively (this is a good implementation for simulations with slotted-time management). If time management is not slotted, an equivalent choice is to uniformly generate direction θ and speed s to be maintained for a constant distance *ds* and *dd,* respectively. This can be a good choice when the system to be modeled has underlying grid topologies, for example, cells, and every single cell migration is a relevant event. Given a similar choice, the "next move" events are not synchronous in the simulation. This can result in additional problems and computation needs: if it is required to obtain the position of neighbor MHs before every move, then every neighbor MHs' position would need to be interpolated. Dealing with the simulated area boundaries, if the area limit is "bouncing" off the MHs, e.g., with a direction proportional to the angle of incidence, in 1-D and 2-D there is an interesting property: every MH will randomly move around its initial position [13]. This also represents a pitfall for this model: if the initial allocation of MHs is not uniform, and the average speed is low, then the "memory effect" of the initial distribution would be persistent, and clusters of MHs could be maintained, despite the randomness of the motion model. This behavior may affect the assumptions about the average degree (i.e., number of neighbors) of a mobile host during the simulation. If the time values *ts* and *td* are long, given the bouncing behavior of area boundaries, the average distribution of MHs would be concentrated in the middle of the simulated area (because when a mobile host is near the boundaries, it has a high probability to be reflected or to choose a new direction toward the center of the area) [3, 13]. If *ts* or the average speed is great, the node density could be made quite uniform with a torus border area policy [3]. Many additional choices or assumptions can be made for this model. The main factors to define for analysis based on this model are the speed ranges and average speed values. The speed factor defines how far a mobile host can roam away from its initial position, and should be dependent on what is intended to be simulated (e.g., micromobility within one room, macromobility between cells, etc.) and on time management factors as well, such as slot duration, for instance. Any biased distribution for speed and direction might lead to a different implementation of the model. Special attention is required regarding the MHs' density assumption, and initial distribution of MHs' positions [13, 54].

- *Restricted Random Model.* The restricted random model usually introduces some kind of autocorrelation or biasing in the "randomness" of the uniform distribution of random model parameters [13]. For example, the speed selection s or the direction θ can be updated up to a limited amount based on current values, $s \in [s - k, s + k]$, $\theta \in [\theta - \pi/4, \theta + \pi/4]$. This model defines a preferred direction and a preferred speed range for all users (or for every single user), and smoothed curves and accelerations. The main factors to define for analysis based on this model are the direction and speed ranges, and the admitted tolerance for variations.
- *Smooth Random Mobility Model.* This was proposed in [3, 5] and can be seen as an extended Random Mobility Model. It is defined with two stochastic processes for correlated speed and direction management. In [28], a criticism of the random models used for MANET simulation was based on the unrealistic movement behavior caused by sudden and uncorrelated speed and direction changes. Restricted random models introduce autocorrelation. In the Smooth Random Model, correlation is introduced together with a set of tunable parameters concerning "node classes," characterized by acceleration and deceleration parameters, target speed, and smoothed direction changes. The proposed model is able to implement realistic behavior of nodes in many scenarios, from urban (Manhattan-like) to large-scale, with acceptable additional computation required [3].
- *Random Waypoint Model.* This is similar to the Random Mobility Model, but it adds the *epoch* and *pause* concepts to make the random model a little bit more similar to realistic user mobility [4, 12]. A MH executes a sequence of epochs, each one defined as a motion interval followed by a pause interval. At the beginning of a motion interval, the MH selects the new destination coordinates (x, y) (not the direction as in the Random Mobility Model), uniformly distributed in the simulated area. Any border policy is equivalent with this motion model, since MHs can only touch, never hit, the area boundaries. Speed is uniformly distributed in [*minSpeed, maxSpeed*]. At the end of a motion interval, a given "pause time" *pt* is defined, uniformly distributed in [0, *maxPauseTime*]. If $pt = 0$, something really similar to the Random Mobility Model is obtained. Intuitively, this motion model is the behavior of the "walking philosophers" (walk, think). This model is of widespread use in many simulations of wireless mobile systems [4, 12, 64]. All of the considerations regarding the Random Mobility Model are still valid, for example, uncorrelated and unpredictable mobility, and memoryless property for speed and direction [4]. The pitfall

of MHs' concentration in the center of the simulated area is still present. This means that MHs are often moving toward the high-density center of the simulated area, and sometimes move temporarily to the sparse boundary areas [5, 54]. Any initial distribution of MHs is not relevant for the steady-state distribution of MHs, because the next position is always a random point in the simulated area. Transient effects from the initial distribution of nodes can be quickly eliminated, in order to avoid biasing in the steady-state simulation results. Some problems can be caused by the model factors: speed and pause time. Currently, Random Waypoint is subject to criticism [64], mainly for the speed distribution of nodes and for the risk of density concentration of MHs in the center of the simulated area [54]. A nontrivial relationship between average speed and average pause time has been reported in many scenarios, depending on the objective of the simulation [13]. If the network stability and link reliability are under analysis, the average pause time sometimes has a prevailing effect with respect to average speed of MHs [13].

- *Random Direction Model.* This is a small variation of Random Waypoint epochs, defined in order to avoid the MH concentration in the center of the simulated area. To obtain a uniform number of neighbors (i.e., degree) for each MH, the modeler should be careful about the model parameters. The model is similar to Random Waypoint: before a motion period, a speed and a direction (as in the Random Mobility Model) is uniformly selected, to be maintained up to the area boundaries will be reached [54]. Once on the boundary, a pause time is selected, then a new epoch starts. Given the reduced density distribution, network partitions are more probable in this model [13]. Another variation is the *Modified Random Direction Model,* in which the selected direction is followed up to a given distance *d,* without necessarily reaching the area boundaries. This model would be quite similar to a hybrid Random Mobility Model with pause times, and to the Random Waypoint Model.
- *Boundless Simulation Model.* This is similar to a vector-based implementation of the Restricted Random Mobility Model, implemented over a torus-like simulation area [13].
- *Gauss–Markov Mobility Model.* This model uses a tuning parameter $\alpha \in [0, 1]$ to vary the degree of "randomness" and self-correlation of speed and direction in a Random Mobility Model [13]. $\alpha = 0$ returns a Random Mobility Model, whereas $\alpha = 1$ returns a linear motion in the initial direction and speed [13].
- *Mobility Vector Model.* This model uses a base vector, a deviation vector, and an acceleration parameter α to define the mobility vector for every MH. Given the mobility vector definition, the extension to a 3-D space model is straightforward, and the vector model defined can be considered as a framework for many models' implementation [27].
- *City Section Mobility Model.* This is a hybrid model merging the Random Waypoint Model and Manhattan-like scenarios. The urban constraints are defined as usual (streets, one-ways, crossings, walls, etc.). Every MH randomly selects a destination, then it travels towards the destination by following the most linear route. Once arrived at its destination, the MH pauses for a random time, then it chooses another destination [13]. The model may introduce some additional issues to be managed, like speed limits, traffic lights, and traffic laws. This may require a significant computation.
- *Graph-Based Mobility Model.* This model has some similarities with the City Section Model. Every MH moves following the edges of a graph defining the infra-

structure of the area. The target destination is one vertex of the graph, randomly selected, and the route is always the shortest path [60].

- *Random (Manhattan) Drunk Mobility Model.* This is similar to the City Section Mobility Model, but it does not define a target point to reach. Every time a new crossing is reached, a new direction is selected from the available ones, according to any distribution probability. Speed can be changed as a separated stochastic process, or according to scenario constraints [3].

Among the synthetic mobility models, the *group mobility models* belong to a new class of models that can be used for MANET modeling and simulation purposes [13, 28, 61]. The main difference for such models is given by the idea that MHs' decisions about their movements would mainly depend upon other MHs in their group or common factors in the scenario. This introduces a motion-correlation effect among MHs belonging to the same logical group. This effect should be evaluated as unacceptable if the assumption for the analysis requires uncorrelated mobility. It may happen, for example, that the relative mobility of MHs within the same group is really low, thereby favoring intragroup communication and routing. The analysis of a given routing protocol under this mobility model should not be considered as a generalized result for general scenarios, because it is biased in a significant way by the adopted mobility model correlation. Group partition and definition is out of the scope of this presentation. It may be related to position, host speed (walk, car, train), and scenario characteristics (e.g., highway lane). The group mobility models can be roughly classified as gravity models, location-dependent models, targeting models, and random group mobility models [13]:

- *Gravity Model.* This model can be used in scenarios where MHs may tend to move toward some destinations (e.g., signal sources) named *attraction points*. Intuitively, every MH is assigned a positive charge, and attraction points are assigned a negative charge. Opposite charges attract each other, while same charges repel each other. MHs with no charge have no gravity effects [13, 27].
- *Reference Point Group Mobility (RPGM) Model.* This is the most general group mobility model. Specifically, the Column Model, Nomadic Community Model, and Pursue Model can be implemented as special cases of the RPGM model [28]. A logical center for the group is defined, and each MH defines a reference point fixed with respect to the group's logical center. The logical center moves according to a group's motion vector (GMV), randomly chosen or predefined, and every MH adds a random motion vector (RMV) to its reference point [28].
- *Reference Velocity Group Mobility (RVGM) Model.* This model can be used when the group shares velocity and direction characteristics, rather than proximity [61]. A group velocity vector defines the dominant velocity characteristic of the group, and a random local velocity deviation vector is composed with the group velocity to determine the single host velocity vector. This can be thought of as the time derivative of the position-based group representation in the RPGM model [61].
- *Exponential Correlated Random Mobility (ECRM) Model.* This model introduces a quite complex motion function that can be used to define the MH movements, where a parameter τ defines the mobility factor, and a random Gaussian variable with parameter σ is included in the formula [28]. The main problem with this model is to find appropriate values for the model parameters.

- *Column Model.* This model defines a mobility pattern similar to a column of not-well-trained soldiers marching in line. Every MH has a reference point in the column and moves randomly around that point. All reference points (i.e., the column) move together based on the common advance vector definition [13].
- *Nomadic Community Model.* This model defines a mobility pattern similar to a group of students on a guided visit to a museum. It is a hybrid random-/targeting-group mobility model. The whole group of MHs (students) has a common single reference point (the guide), which is moving according to a given random mobility model (e.g., similar to Random Waypoint). Every single MH is free to move around the group's reference point, according to a random mobility model [13].
- *Pursue Model.* This is another targeting-group mobility model based on the definition of a single target moving according to a random mobility model. The tracking MHs define their group mobility based on the straight direction from their position to the target, biased by a random offset vector.

Other complex and mathematically intractable motion models can be defined to capture more realistic user mobility patterns to be used in simulation. This is an ongoing research activity. One of the challenges for the research is to find efficient techniques for the implementations of the proposed models. Additional efforts should be made to study models whose implementation can be supported efficiently in the adoption of the distributed simulation paradigm.

Many commercial and freely distributed simulation tools support mobility models and complex scenarios. Recently, some application tools have been proposed for known simulation tools and models. CAD-HOC [55] is a tool used to generate mobility benchmarks and ad hoc scenarios to feed the network simulator ns-2 [45]. Bonn-Motion is a mobility, scenario-generation, and analysis tool, written in Java, that can be used to define Tcl scripts feeding ns-2. FraSiMo [20] is a research project to model mobile ad hoc networks with Omnet++ [46]. A commercial tool, OPNET [47], defines a complete set of facilities to model complex mobility scenarios and propagation models for ad hoc networks.

14.2.1.6 Traffic Workload.

The workload characterization for MANETs, that is, the amount of data to be transmitted between MHs, is another relevant point for the modeling definition. Workload is relevant for the evaluation of the supported Quality of Service (QoS) and service reliability for the application and user needs. The network traffic characterization is a problem that has been analyzed for years, dealing with the self-similarity, bursty nature, and correlation of packet-arrival processes, among other things [56].

Trace-based workload models are widely used in many simulations, data and video transmission, for instance. In MANETs, currently nobody knows what would be the killer application, so we can only speculate about the workload characterization of such systems. Usually, as a worst-case scenario, the simulation analysis can be performed under *asymptotic workload conditions.* This means that the assumption for the system is that the sources of traffic in the network always have full transmission buffers. This hypothesis is good for testing the stability and congestion reaction of a given network, or to evaluate the scalable behavior and asymptotical throughput metrics for the system.[2]

[2] Note that this is a worst-case scenario, and maybe an unrealistic condition.

Underload conditions can be defined by adopting other commonly used, parameterized models. Another widely adopted traffic model for MANET simulation analysis is the *Constant Bit Rate (CBR) Traffic Model,* in which every source (sender) of traffic generates a constant flow of packets. This can be obtained simply by assuming that a given amount of data is generated at constant time intervals. This model is commonly used to approximate the workload generated by voice-based applications. This model can also be extended in many ways, in order to make it much more realistic.

The *Variable Bit Rate (VBR) Traffic Model* can be adopted to approximate the workload generated by data and video applications [41]. It is defined by traffic sources generating a variable amount of data, as a function of time, depending on many packet-interarrival-distribution parameters.

14.2.2 Mobile Ad Hoc Network Simulation

Computer-based discrete-event simulation is one of the most flexible methods for the performance evaluation of complex systems such as MANETs. The goal of a simulation study is the construction of a simulator that mimics the system state transitions and, by collecting and analyzing data during simulation runs, estimates the performance metrics of the systems under analysis. An orthodox simulation study is based on several steps whose characteristics and number can vary with respect to the nature of the system analyzed and the objectives of the study. The key steps in establishing the kernel of any simulation study are (1) problem formulation, (2) workload characterization, (3) model definition and validation, (4) construction and verification of the simulator, (5) design of experiments, and (6) analysis of the simulation results or output analysis [29].

In this section, we discuss the main system characteristics and performance figures of interest for mobile ad hoc network simulations. Regardless of the applications and the protocol layers considered for the analysis, many critical features contribute to determine the efficiency, reliability, and effectiveness of MANETs. MANET networks are characterized by dynamic topologies, requiring adaptive, multihop routing protocols, dealing with bidirectional and unidirectional links [18]. Links are bandwidth constrained compared to typical wired networks, and they offer variable capacity and delay times, due to the effect of highly variable scenario conditions. Mobile hosts are energy constrained, so MANETs privilege protocols dealing with energy reduction approaches, for example, sleep-period management and adaptive power reduction. Due to host mobility, MANETs' scalability is a major problem to solve. This problem is complicated further by the distributed management and distributed protocols' implementation, which are commonly adopted [24]. This makes it difficult to guarantee network behavior, reliability, fairness and efficiency under every condition. Reduction of overheads to maintain proper network functionality is a common problem: use of critical resources, like battery energy, buffer memory, local CPU computation, and bandwidth for the transmission of control packets should be minimized.

14.2.2.1 Performance Metrics. A large set of *performance metrics* could be defined to evaluate MANETs, in order to understand the critical features of the considered system. Some metrics can be considered relevant or significant only for a given protocol layer. Other metrics can be general, even if they may be affected by a chain of interlayer implications. Performance metrics can be roughly divided into the following three categories: user-performance metrics, resource-utilization metrics, and system metrics. User-performance metrics include, but are not limited to, latency, delay, quality of service, priorities, average and peak performance, reliability, and cost-efficiency metrics. Resource utilization metrics include overheads, utilization, fairness, and efficiency, just to mention a few. System metrics include scenario, stability, scalability, and context metrics (e.g., topology changes, network partitions, cluster life, mobility, density, load, path length, etc.). As an example, given a task-process evaluation, interesting metrics can be defined as average power consumed and communication overheads (which are both considered as be resource utilization metrics) and task completion time (i.e., a user-satisfaction parameter). Given a routing protocol evaluation, interesting metrics can be defined as average end-to-end throughput, average end-to-end delay, average link utilization, average packet-loss probability, energy efficiency, and protocol overheads, among other indices.

Now we will present a short list of generalized metrics that can be evaluated and adopted in the analysis of Medium Access Control, Routing, and Transport protocols for MANETs [24, 18, 23]. In the analysis of the following metrics, mean values should be investigated together with variances or confidence intervals and distribution percentiles.

- *Access Delay.* This is the time spent by a frame (or a packet) in the MAC (routing or transport level) queue. It is defined from the instant the frame is queued (or dequeued) until its transmission is successfully completed. Since delay depends on protocol definition and also on system load and traffic model, every comparison should be performed under the same conditions.
- *Channel Capacity.* This is the maximum amount of data (e.g., bit rate C) that can be transmitted over a single channel. The nominal bit rate can be reduced in the presence of noise and interference. Coding techniques scale in the number of bits/symbols in order to contrast the noise, resulting in lower and lower bit rates.
- *Throughput and Utilization.* The scope of any transmission protocol is to maximize the number of transmitted bits while minimizing the average access delay. Throughput T is defined as the average size S of a given frame (packet), divided by the corresponding average access delay D, that is, $T = S/D$. This index is related to the utilization index U, which can be defined as the fraction of channel capacity C used for successful data transmission.
- *Overheads.* Every resource in the system that would not be strictly necessary to transmit the payload of the communication can be considered as an overhead and should be minimized (e.g., time, bandwidth, capacity, CPU time, energy, money).
- *Fairness.* This is a concept related to service and resource sharing, rather than a performance index. A transmission protocol is fair if it does not show any preference for any single MH contending or waiting for resources or services. Fairness is the opposite of prioritized access and scheduling policies, adopted to support QoS and multimedia applications.
- *Stability.* A stable system should not have any fluctuating behavior resulting in a reduction of the average throughput and utilization. Adaptive protocols should be evaluated under the stability viewpoint. Many factors contribute to make the system unstable.
- *Reliability.* This concept defines a measure of the system reliability with respect to many failures that can be expected, for example, network partition and broken paths. The reliability can be evaluated as a probability measure of failures, and as a measure of the failure-recovery delay.

- *Scalability.* A scalable system is obtained when protocols and management react and adapt in an opportune way to changes in the system factors like load and number of MHs. A scalable system is a system in which performance scales with no collapse. If a collapse occurs, it would be interesting to find information about the *saturation point,* that is, the limit the system can sustain, and the recovery time from saturation conditions. A typical example is given by congestion problems.
- *Power Consumption.* Most MHs are battery powered, and maximum energy efficiency is required for every task performed, including system maintenance, transmission, and reception of data.

14.3 SIMULATION TECHNIQUES

In this section, we shall introduce the basic terminology and major issues pertaining to simulation techniques. Before, we proceed further, we must draw distinctions between different types of simulations: *continuous, discrete,* and *hybrid.*

Continuous simulation models the situation in which changes in state occur smoothly and continuously in time, for example, the flow of liquid through a pipeline, weather modeling, and circuit-level simulation of electronic components. Continuous simulation models often involve difference or differential equations that represent certain aspects of the system. *Discrete* simulation refers to the modeling technique in which changes to the state of the model can occur only at countable points in time [21, 57]. For example, in logic simulation, the circuit is simulated by assuming that node voltages only take on values from a finite set (say, 0 and 1) and that transitions between values are instantaneous; in switch-level simulation, transistors are simulated as switches that can be either opened or closed. Digital computing, communication, and queueing systems (such as used by bank tellers and job shops) are other examples of discrete event systems. Many systems are *hybrid,* that is, they contain combinations of discrete and continuous characteristics. An example of a hybrid system is an unloading dock where tankers queue up to unload their oil through a pipeline. The decision of whether to use a discrete or continuous model for a particular system depends on the specific objectives of the study. For example, a model of traffic flow on a freeway would be discrete if the characteristics and movement of individual cars were important. Alternatively, if the cars can be treated in the "aggregate," the flow of traffic can be described by differential equations in a continuous model.

In this chapter, we are interested into discrete systems that can be simulated by discrete-event simulations. In a discrete-event simulation, the model evolution is defined by instantaneous *events*. Each event corresponds to a transition in a portion of the model state, composed of *state variables,* each describing a characteristic of the model. Each event also has a simulation time associated with it, called a *timestamp,* which defines its occurrence time. Each event may in turn generate new future events.

The generation of new events and the dependency of their transitions on state variables that previous events may have updated define a relation of *causal order* (a partial order) among events. Related events are said to be *causally dependent,* whereas unrelated ones are called *concurrent*. In order to guarantee the correctness of the simulation, concurrent events may be safely processed in any order in a simulation, whereas causally dependent events must be processed according to the causal order. Thus, to ensure the strict chronological order, events are processed one at a time, resulting in an (apparently) sequential program. A typical template for a sequential simulation is given in Figure 14.5.

```
While Not Empty (EventQueue) Do
        dequeue (m)                /* earliest event from EventQueue */
        update (clock)
        simulate (m)
        enqueue()                  /* enqueue any events produced */
EndWhile
```

Figure 14.5. Basic Sequential Discrete Event Simulation Algorithm.

Discrete systems can be simulated by discrete-event simulations. Many methods have been proposed in the literature for implementing discrete systems. They can be broadly classified into two groups, the *synchronous* and the *asynchronous* methods. In synchronous discrete event simulation, all objects in the simulation progress forward in simulation time together, in synchrony, with no object ahead of any other in time. The usual queue implementations for sequential simulation are all synchronous methods. In contrast, an asynchronous method permits some objects to simulate ahead in time while others lag behind. Of course, an asynchronous method must include some mechanism for ensuring that when an object that is "behind" schedules an event for execution by an object that is "ahead," it does not cause any events to be executed in the wrong order.

In this chapter, we are interested into modeling and simulation of wireless and mobile networks based upon asynchronous discrete event simulation tools.

14.3.1 Sequential Network Simulation Testbeds

In this section, we shall review several network simulators that have been widely used by both academia and industry communities.

OPNET (Optimized Network Engineering Tool) [47] provides a comprehensive set of simulation and modeling products for the development and performance analysis of wired and wireless networks. For protocol and technology R&D space, their OPNET Modeler product provides a high-fidelity discrete-event simulation environment using a hierarchical modeling paradigm, in which each level of hierarchy represents different aspects of the complete model being simulated. It provides powerful tools to assist users in building simulation models, and for output analysis. OPNET includes a highly optimized discrete event simulation kernel, and has been used quite successfully within the wired and wireless networks communities. To the best of our knowledge, scalability is a major problem in most monolithic discrete event simulators, such as ns-2, and takes too long to run the simulation, and one has to spend a large amount of time to understand how to use them. OPNET is among a few commercial products that are reasonably easy to use, and relatively scalable when compared to existing simulators. To further increase its scalability, parallel simulation technology has been under development.

Recently, a federated simulation approach has been investigated to enhance and parallelize OPNET. Each confederate is basically a sequential simulator modeling a subnetwork of the simulated model. Although, recent results were quite encouraging, the work is still at an early stage [63].

INSANE, a network simulator, was designed to test various IP-over-ATM algorithms with realistic traffic load derived from empirical traffic measurements. Although the simulator provides an easy approach to check the progress of multiple running simulation processes, we find it quite restrictive to ATM network simulations.

NetSim is another network simulator. It was designed to provide detailed simulation of the Ethernet, including realistic modeling of signal propagation, collision detection, and handling process.

OMNeT++ [46] is a freely distributed, object-oriented, modular, discrete-event simulator written in C++. It is designed for general-purpose discrete-event simulation, and provides some model libraries for communication protocols and network systems. NED, a network descriptor language, can be used to assist the modeler in the model definition based on system modules written in C++. OMNeT++ support for parallel execution and parallel discrete event simulation is an ongoing research activity.

The network simulator *ns-2* [45] is a discrete-event simulator that provides substantial support for simulation of TCP, routing, and multicast protocols over wired, wireless (local and satellite), and wireless multihop ad hoc networks. *ns-2* began as a variant of the REAL network simulator in 1989 and has evolved substantially over the past few years. Since then, it has included substantial contributions from other researchers, including wireless code from the UCB Daedelus and CMU Monarch projects and Sun Microsystems. *ns-2* is written in C++, and it uses OTcl, an object-oriented version of tcl, as a command and configuration interface. The interface code to the OTcl interpreter is separate from the main simulator, and complex objects are decomposed into simpler components for greater flexibility and composability. Although *ns-2* is widely in use within the wireless networking communities, it is not a fine-tuned and finished product, and it is still a result of an on-going effort of research and development. In particular, bugs in the software are still being discovered and corrected. Users of *ns* are mainly responsible for verifying for themselves that their simulations are not invalidated by bugs.

Among all the existing network simulators, *ns-2* is most popular tool used by both the wireless and wired communities. It has also been extended to mobile ad hoc networks as well. The major drawback of *ns-2* is the execution time of the simulation, mainly due to the sequential implementation of the discrete-event simulator. Although it is quite easy to use, run, or modify preexisting models, it requires a large amount of time to study the inside of *ns-2* before a simulation modeler develops new models.

A few researchers have investigated ways to speed up the running time of the simulation using *ns-2*. We shall describe them in the next section.

14.3.2 Parallel and Distributed Simulation

Due to the enormous computational requirements of a sequential simulator for complex wireless systems, parallel discrete-event simulation techniques [14, 39, 40, 43, 48, 65] are often studied to reduce the execution time of the simulation models. Before, we proceed further, let us introduce the basic terminology and major issues pertaining to parallel and distributed simulation. A parallel or distributed simulation should provide the same solution to a problem as a sequential simulation.

14.3.2.1 Principles of Parallel and Distributed Simulation. To ensure strict chronological order in large-scale simulation, events are processed one at a time, resulting in an (apparently) sequential program. Only by eliminating the event list in its traditional form, so as to capture the interdependence of the process being simulated, can additional parallelism be obtained [16]. This is the objective of *parallel simulation*. Indeed, parallel simulation shows great potential in terms of exploiting the inherent parallelism of a system, and the concurrency among events to achieve execution speedup. Good surveys of the literature may be found in [21].

A parallel simulator is composed of a set of *logical processes* (*LPs*) which interact by means of messages, each carrying an event and its timestamp, thus called *event messages*. Each LP is responsible for managing a subset of the model state, called the *local state*. Each event *e* received by an LP represents a transition in its local state. The events scheduled by the simulation of *e* are sent as event messages to neighboring LPs to be simulated accordingly. In a simulation, events must always be executed in increasing order. Anomalous behavior might result if an event is incorrectly simulated earlier in real time and affects state variables used by subsequent events. In the physical model, this would represent a situation in which future events could influence the present. This is referred to as *causality error*. Several synchronization protocols have been proposed to deal with this problem. These techniques can be classified into two groups: *conservative* and *optimistic*. Conservative synchronization techniques rely on *blocking* to avoid violation of dependence constraints, and *optimistic* methods rely on detecting synchronization errors at run time and on recovery using a *rollback* mechanism.

14.3.2.2 Conservative Simulation. Conservative approaches enforce event causality by requiring that each LP elaborate an event only if it is certain that it will not receive an earlier event. Consequently, events are always executed in chronological order at any LP. Each logical process LP_i maintains an input queue (l_{ij}) for each of its neighbors LP_j. In the case that one or more (input) queues are empty, the LP is blocked because an event with a smaller timestamp than the timestamp of the waiting events might yet arrive at an empty queue. This mechanism implies that only unblocked LPs can execute in parallel. If all the LPs were blocked, the simulation would be deadlocked. Ensuring synchronization and avoiding deadlocks are the central problems in the conservative approach. Several schemes have been proposed to alleviate this problem. In [16], the authors employ null messages in order to avoid deadlocks and to increase the parallelism of the simulation. When an event is sent on an output link, a null message bearing the same timestamp as the event message is sent on all other output links. As is well known, it is possible to generate an inordinate number of null messages under this scheme, nullifying any performance gain [21].

As a result, a number of attempts to optimize this basic scheme have appeared in the literature. For example, in [57], the authors refrain from sending null messages until such time as the LP becomes blocked. They refer to this approach as *eager* events and *lazy null* messages. They reported some success in using variations of Chandy–Misra approaches to speed up logic simulation.

Boukerche and Tropper [10] employed the following approach. In the event that a null message is queued at an LP and a subsequent message (either null or event) arrives on the same channel, they overwrite the (old) null message with the new message. A single buffer is associated with each input channel at an LP to store null messages, thereby saving space as well as the time required to perform the queuing and dequeuing operations associated with null messages. Good surveys of conservative techniques can be found in [11, 21].

14.3.2.3 Optimistic Approach. Time Warp is based on an optimistic approach and enforces the causal order among events as follows. Events are greedily simulated in timestamp order until no event messages remain or until a message arrives in the “past” (a straggler). Upon receiving a straggler, the process execution is interrupted, and a *rollback* action takes place using *anti-messages*. Each message is given a sign; positive messages indicate ordinary events, whereas negative messages indicate the need to *retract* any corresponding event that was previously processed. Similar messages that have different signs are called anti-messages. If a negative message is received, the message and the corresponding anti-message are both *annihilated*. A rollback consists of the following three phases:

1. *Restoration.* The latest state (with respect to simulation time) valid before the straggler's timestamp replaces the current state, and successive states are discarded from the state queue.
2. *Cancellation.* The negative copies of messages that were produced at simulation times successive to the straggler's timestamp are sent to the proper processes, to possibly activate rollbacks there.
3. *Coasting-forward.* The effective state that is valid at the straggler's timestamp is computed by starting from the restored state and by elaborating those messages with a timestamp up to the stragglers; during this phase no message is produced.

Rollbacks are made possible by means of *state checkpointing*. The whole state of the process is checkpointed into the *state queue* according to some discipline [49].

To minimize the storage overhead required to perform rollbacks, and to detect the termination of LPs, optimistic synchronization mechanism uses a *local virtual time* (*LVT*) and a *global virtual time* (*GVT*). LVT represents the timestamp of the latest processed event at an LP, whereas GVT is defined as the minimum of all the local virtual times of all LPs, and of all the timestamps of messages in transit within the simulation model. GVT indicates the minimum simulation time at which a causal violation may occur. GVT computation is used to *commit* the safe portion of the simulation.

The optimistic scheme is preferred when a system to be simulated contains high predictability of events so that rollbacks are kept to a minimum. Thus, to improve the PCS network simulation, we use a hybrid approach with both conservative and optimistic schemes.

14.3.3 Wireless Network Simulators Based upon PDES

Several simulation techniques have been proposed in the literature [14, 38, 39, 43, 48, 65] to speed up the execution of simulation of large-scale wireless networks. In this section, we shall describe them, and discuss their main features.

ns-2 has long been considered to be a de facto standard simulator for wireless and wired networking protocols research. The networking community has long been resistive to rewrite the network simulator or use different platforms. Therefore, some researchers have tried to parallelize the *ns-2* using established parallel and distributed simulation techniques, thereby providing a transparent parallel execution of *ns-2*. Riley and Fujimoto have proposed a distributed version of the popular network simulator, *ns-2,* which they refer to as parallel and distributed *ns,* or simply PDNS [50]. Their goal is to use the existing network simulator and minimize the changes to it while allowing their parallel simulator to take advantage of their proposed new version of *ns-2*. They revised the *ns-2* syntax by adding a set of directives that are directly related to the parallelization of the simulation.

Their idea is based upon the federated simulation approach, in which separate subnetworks of the simulated model are executed on different processors connected either via a Myrinet network, or a standard Ethernet network using the TCP/IP protocol stack. A library, which they refer to as Georgia Tech RTI Kit [25], is used for synchronization purposes. Conservative methods for synchronizing *ns-2* processes have been implemented. The RTI Kit is a software implementation of the Run-Time Infrastructure of the Department of Defense's High-Level Architecture (HLA) for large-scale distributed simulations [19]. Much of the improvement obtained in their design was obtained from parallelization of the setup of the simulation and not the actual execution of the simulation.

Another project at the University of Cincinnati [17, 33] involved running *ns-2* in parallel. The main objectives of this work is to build a space–time parallel simulator to study how effectively ad hoc network simulations can be performed in parallel. At the present time, their testbed supports parallel execution of *ns-2* programs consisting of point-to-point links with static routing and UDP traffic. A conservative null-messages approach has been used for synchronization purposes. Although initial results are encouraging, this work is still at an early stage.

The PDES community has also tried to design efficient simulators for wireless and mobile systems using PDES synchronization schemes without relying on preexisting network simulators. Wireless Propagation and Protocol Testbed (Wippet) [48] is a versatile simulator for wireless networks. It consists of basic set of modules implemented using the TeD, an object-oriented and telecommunication-descriptive language for parallel simulation of telecommunications developed at Georgia Tech. [51]. Its propagation and interference modeling at the receiver MH made simulation suitable for studying dynamic channel-allocation schemes. The partitioning of the model into multiple zones is either geographically based or channel based. Channel-based partitioning gives rise to better speedup due to the rare synchronization of zones in which a mobile device changes the channel. However, how that is achieved in the implementation of Wippet is unclear. Selection of other channels requires interference measurement on the destination channel, which should induce overall synchronization on all zones.

The *GloMoSim* (Global Mobile Information System Simulator) is a library-based sequential and parallel simulator for wireless networks, including multihop wireless ad hoc networking and traditional wired Internet connectivity [58, 65]. The GloMoSim is designed as a set of library modules, each of which simulates a specific wireless communication protocol within the protocol stack. Modules of the protocol can be developed at different levels of granularity. It has been developed using PARSEC (Parallel Simulation Environment for Complex Systems), a C-based parallel simulation language [1]. PARSEC basically adopts a message-based approach to discrete event simulation in which physical processes are modeled by simulation objects referred to as entities, and events that represent the transmission of timestamped messages among corresponding entities. Glomosim has been designed so that it can be easily extended and new protocols and modules can be added to this library using this PARSEC language. It has been implemented on both shared- and distributed-memory machines, and it supports conservative layered simulation in the context of wireless network simulation. The synchronization protocol makes use of Chandy–Misra null-messages scheme [16, 57]. Although the results reported in [65] show a significant reduction of the null-messages overhead, a speedup of only up to 3.5 was obtained using eight processors [65]. Low speedups hinder the improvement due to unresolved causal dependencies. More recently, the authors have reported improvement on conservative simulation due to better lookahead computation [43].

Both GloMoSim/Parsec and TeD/GTW systems require the simulation modelers to learn new language extensions to describe their network models, although GloMoSim, as opposed to TeD/GTW, was designed to make the mechanics of parallel simulation transparent to protocol modelers by embedding them into the lowest (channel) layer. Further knowledge of PDES synchronizations is needed to understand how they work, in order to develop new models, unless users are interested in running or modifying preexisting models.

QualNet is basically a commercial product derived from GloMoSim, developed at UCLA. It is designed by the Scalable Network Technologies Inc., headed by R. Bagrodia from UCLA. Several extensions have been added to GloMoSim to facilitate the development of new protocols for wired, wireless, and ad hoc networks.

The following summarizes other related work on parallel simulation of wireless networks. An optimistic model based on Time Warp is proposed in [14]. It uses logical processes (LPs) and uniform rectangular-shaped cells to simulate large-scale PCS networks. The mobility of a MH is limited to four neighbors only, and low blocking probability is achieved with a fixed ratio of 50 MHs per cell. Better results were obtained with a low number of mobile hosts per cell. In [39], another optimistic parallel simulation is presented in which the PCS coverage area is modeled by fixed hexagonal shaped cells identifying the LP. The MHs are given constant speed and angle of movement. Although the obtained results are encouraging, the model is useful only for low call traffic and reduced mobility.

As opposed to the preceding two-cell-based partitioning, a channel-based partitioning is proposed in [40]. In this method, when a MH makes a hand-off, a set of messages is sent out to all channels available on the new BS. This scheme may generate an inordinate number of messages, nullifying any performance gain. Mobility of MHs is limited to constant speed and four directions—north, east, west, and south. The MH is disposed after the call is terminated. A good analysis of break-even points between cell-based and channel-based partitioning were reported.

Despite the fact that promising results were obtained in these approaches, most of them ignored real-life patterns for mobility and PCS network deployment by restricting cell shapes to uniform geometric objects such as hexagons, rectangles, or squares. These limitations and weak spatial modeling of cell characteristics often simplify the simulation model, and do not capture the accuracy and realism in PCS networks performance evaluation. Linear movements have been used in some of the existing works [14, 39, 40, 43]. In real transportation traffic flow, segmented movement patterns, occasional pauses, and, most importantly, rush hour traffic and/or congested roads, trigger spikes in the call arrival rate. The results reported in [14, 39] assumed a (fixed) ratio of MHs to channels per cell, which is unrealistic. Furthermore, channel-based partitioning [40] creates an MH at runtime and discards it after the call is terminated, thus losing mobility within calls and requiring unrealistic call-arrival processes that may be unrelated to mobility patterns.

In [6], Bononi et al. have recently defined a prototype General Adaptive Interaction Architecture (GAIA) middleware to be implemented over a conservative, HLA-based, distributed simulation of mobile systems. The aim of the proposed middleware is to provide adaptive runtime allocation of model components over the set of federates executed on the available set of execution units. The adaptive allocation is performed in order to balance the need for parallel execution and the message-passing overhead of distributed simulation. The leading assumption of this work is that mobility inside the simulation model maps on dynamic changes in the area of influence of every simulated host. If a certain amount of time-locality is present in the communication with the neighbor hosts, then adaptive allocation can reduce the amount of inter-federate synchronization-message overheads. Preliminary results show that speedup can be obtained in HLA-based, conservative simulation of mobile ad hoc networks, executed over networked clusters of personal computers.

Recently, Boukerche et al. [7] developed *SWiMNET,* a high-performance simulator for wireless and mobile networks. Their scheme uses a hybrid approach to simulating wireless and mobile networks, based on a combination of optimistic and conservative techniques. It exploits event precomputation due to a simple assumption: *mobility and call arrival events of MHs are independent from the state of the wireless PCS simulation*. Thus, all events for each MH can be precomputed assuming all channel requests are satisfied, and the actual channel allocation simulation cancels events for blocked calls. An exception to this fact may be a *hotspot,* that is, congestion due to rush-hour traffic or at a traffic junction. In this situation, the MHs in that region have very low, if any, mobility and tend to make more calls. This is tackled in the mobility design by introducing hotspot areas where speeds are reduced and call arrival rates are increased.

With this mechanism, all movement and call-related events for each MH are precomputed, assuming all channel requests are satisfied. The small portion of events to be retracted due to blocked calls is later computed in the actual simulation. The low percentage of blocked calls desirable for wireless networks is exploited by the optimistic portion. Event cancellations are done only if a call is blocked or dropped. The precomputation can be pipelined to the channel-allocation simulation, thus minimizing the overhead of generating events.

In Table 14.1, we summarize a comparison of model-related and simulation-related issues for SWiMNet, Wippet, and GloMoSim.

Table 14.1. Comparison of Model and Simulation Issues

	Model-Related Issues		
Parameters	SWiMNet	Wippet	GloMoSim
Mobility	Segmented paths	Manhattan-style	Unspecified
Call arrivals	Poisson process per MH	Model-wise poisson process	Node-wise poisson process
Coverage map	Irregular cells over Voronoi diagrams	Manhattan-style urban environment	Uniform geometry (hexagons or squares)
Signal propagation	Not employed	Stochastic fading	Free-space model
Call admission	FCA	RSSI-based DCA	Unspecified
Handoff mechanism	Cell crossing induced	RSSI based	Cell crossing induced
Model size	54 BSs, 10000 MHs	48 BSs	2000 nodes (short range)
	Simulation-Related Issues		
Precomputation	Mobility, calls	NA	NA
Synchronization	Hybrid	Optimistic	Conservative
Partitioning	Cell based	Zone based (channel/cell)	Static node based
Call traffic	4 calls/MH/hr	6 calls/sec to system	1 pkt/sec to each node
Speedup	11.8 on 16 processors	4 on 8 processors	6 on 16 processors

In what follows, we shall describe the main features of SWiMNet, a recently developed scalable simulation testbed for wireless and mobile networks.

14.3.3.1 Description of* SWiMNet *Model Components. In SWiMNet, the entire simulation model is the result of the composition of four model components: (1) mobility models, (2) call process, (3) BS deployment, and (4) channel management scheme. Although the first three model components are independent of each other, the channel management component is dependent on the compound result of the first three components. Mobilities and calls are represented by independent and stochastic processes[3]: locations of MHs are chosen pseudorandomly, MHs trajectories across the map are sequences of pseudorandomly generated segmented movements, and call interarrivals and durations are pseudorandomly distributed.

As part of the mobility model, the population of mobile hosts (MHs) are classified into groups of *workers, wanderers, travelers,* or *static users,* so as to represent behavior of different users across the wireless coverage area. The number of MHs per class is arbitrary. Movements are modeled such that a complete path is composed of any number of straight segments. This allows almost any kind of movement, by approximating a curve line with as many segments as required by the resolution considerations. Every segment is then logically partitioned into unitary tracts of a given unitary resolution, which defines how finely the MH position is checked.

The call model is specified by means of a maximum call rate per hour per MH, and an average call duration. The entire simulation time interval can be partitioned into any number of subintervals, each with a different call rate. Thus, it is possible to represent call rate changes during the simulation; night hours may be represented with very low call rates, office hours with high call rates.

By composing mobility, calls, and BS deployment, the precomputation stage (Stage 1) is able to generate one stream of possible events per MH. The destination of such events within Stage 2 is precomputed as well. The actual set of possible events, their correlations, and how they are simulated, depends on the channel management policy to be simulated.

The general structure of the simulator consists of three logical levels and two physical stages. Entities comprising logical Level 1 are organized into Stage 1 of n_1 (container) processes, where each process maintains a set of mobile host incarnations (all sets are disjoint). Stage 2 consists of n_2 cell container processes, and implements Level 2 (event sorters) and Level 3 (cell incarnations) of the logical structure. The event sorter and the cell incarnation related to the same cell are managed by the same cell container process in Stage 2. Therefore, communications between Level 2 and Level 3 are easier and faster by means of direct memory access instead of message passing. Communications between Stage 1 and Stage 2 are implemented by means of a Message Passing Interface (MPI) using the LAM 6.1 environment [44]. Since no feedback is necessary from Stage 2 to Stage 1, in principle the execution of the two stages may be performed at different times. However, that would require Stage 1 to store precomputed events on file and Stage 2 to read them from file afterwards, thus adding overhead.

The structure of SWiMNet simulator is depicted in Figure 14.6. For simplicity, every process is represented as composed of only one entity (i.e., MH/Cell objects) per level. Communications between Stage 1 and Stage 2 are based on a conservative scheme using

[3] Note that a discrete distribution with one element only corresponds to a deterministic behavior.

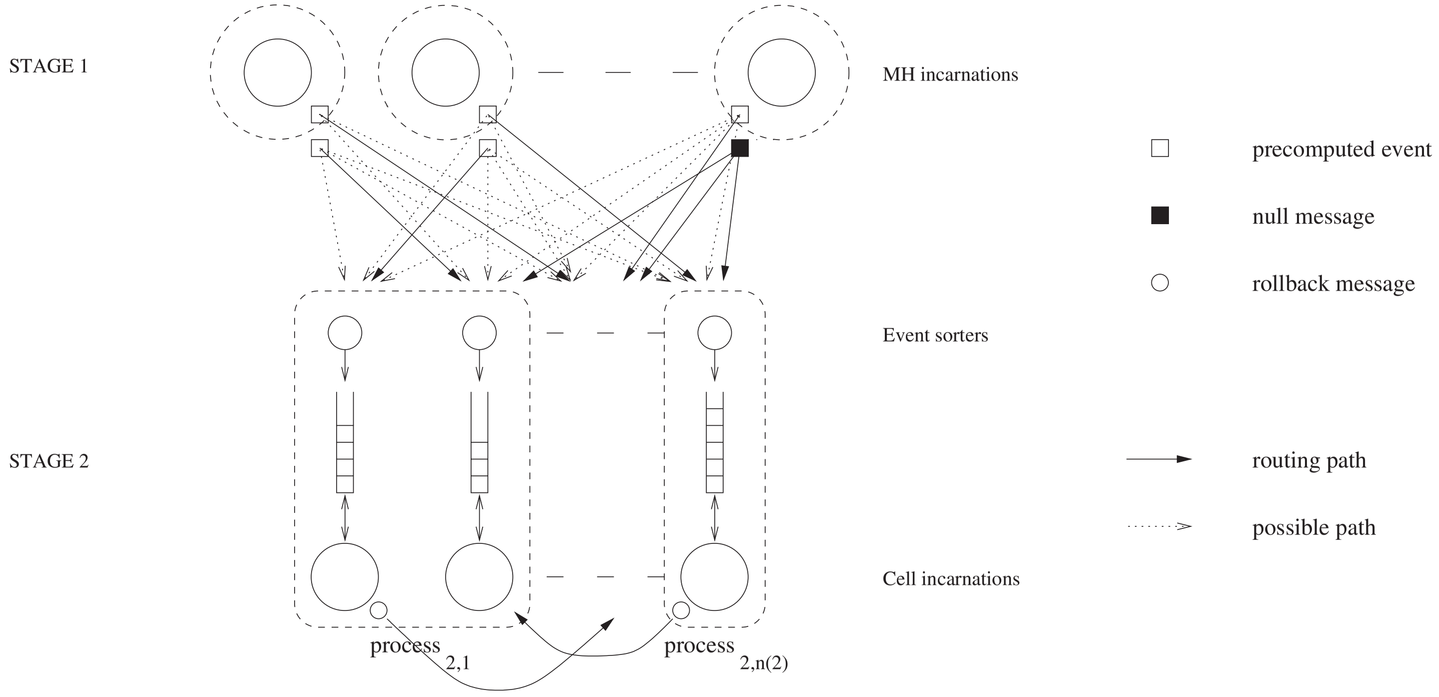


Figure 14.6 Logical interconnection between levels of the parallel simulator.

the null-messages paradigm, whereas communications within Stage 2 are based on an optimistic scheme [30].

The implementation is based on an object-oriented methodology using C++ as the programming language, which makes it easy to maintain and flexible to any changes. Every MH object is constructed as a base class with parameters that are generalized to all mobile hosts. The four classes of mobiles (workers, static users, wanderers, and travelers) are derived from the base MH class. MH objects are created in Stage 1 container processes, whereas cell objects are created in Stage 2 container processes.

Three levels in the SWiMNet simulator have been defined, in which logical activities of objects are elaborated:

1. Movement precomputation (Level 1) is composed of N_{mb} *mobile host incarnations*.
2. Event sorting (Level 2) is composed of N_{cells} *event sorters*.
3. Channel allocation simulation (Level 3) is composed of N_{cells} *cell incarnations*.

The communications within and between the two stages are shown in Figure 14.6. Every MH incarnation process generates events for each MH it maintains. Then, those events are sent to the event sorter process (ES) for the cell where the event takes place. The MH incarnations are independent of each other. Thus, activities of Level 1 are completely parallelizable. Similarly, event sorters are independent of each other. However, cell incarnations, where channel management simulation takes place, are mutually dependent.

In SWiMNet, the optimism lies in the following: every time a move-in event is simulated at any cell incarnation, the latter optimistically assumes that the call is still on, unless it already received notification that the call was blocked by means of a *call blocked* message. In the case in which a late notification is received, that is, a call blocked message is received after the corresponding move-in event has been simulated, a rollback is performed that retracts and corrects the simulations that follow and include the move-in event. A call blocked message is sent by any cell incarnation whenever a move-out event is simulated, if the simulation of the corresponding channel request event in the event couple did not actually result in a channel being allocated. This implies that from Level 1, it is necessary to keep track of at least the event that follows every move-out event in the sequence of events for the same mobile host. This information may then be used to construct a call blocked message.

Rolling back the computation might result in the need for *retracting* an incorrectly sent call blocked message by means of a *call allocated* message. For instance, let us assume that a call was notified as blocked because of channel unavailability after a channel was allocated to another call. If the allocated call turns out to be already blocked in a previous cell, then the call that was notified as blocked can indeed be allocated. A call blocked and a call allocated message relative to the same call correspond to a pair of antimessages in *Time Warp* simulations [30, 31], hence a *sign* can be associated with these messages. In a Time Warp simulator, a call blocked message has a positive sign, whereas a call allocated message has a negative sign.

Due to the optimistic assumption, the elaboration of a simulation message whose corresponding precomputed event was already elaborated always causes a rollback. However, the way in which data are stored in the simulator allows rollbacks to be optimized, i.e., only a portion of precomputed events need to be involved in the rollback, and such a portion is exactly computed without the need to inspect any additional event.

The simulator is automatically generated by the master process, which reads the description file of the simulation model and partitions the model into two stages. The description file includes the mobility model, the call arrival model, the cellular system map, the system architecture, the experiment seed, and the simulation time of termination. The main tasks of the master are (1) generating parameters for every single logical entity, i.e., positions, speeds, movement times, etc. for each mobile host incarnation, according to the general description of the mobility model; and (2) deciding the number of processes per stage and mapping the processes to processors. Process mapping is an important factor in improving the efficiency of parallel simulation protocols. Currently, a simple static mapping discipline has been adopted: given a number of processors allocated to the simulation, half of the processors are used for Stage 1 and half for Stage 2.

Performance analysis of SWiMNet applied to wireless and mobile system can be found in [7, 9]. Further work is underway to evaluate its performance for mobile ad hoc networks.

14.4 CONCLUSION

This chapter focuses on several challenging design and modeling aspects of wireless, mobile, and ad hoc networks. We presented a discussion of modeling issues related to physical transmission and interference, topology, mobility, workload, and performance figures for mobile ad hoc networks simulation.

Modeling and simulation are traditional methods to evaluate large-scale wireless and multihop network designs. However, modeling is often intractable with today's large and complex mobility and traffic patterns in wireless and multihop systems. Thus, researchers have turned increasingly to the use of simulation studies of these systems. Though, detailed simulations of large-scale, wireless, mobile, and ad hoc networks require enormous execution time and large amounts of memory due to the complexity involved in the simulation and mobility models. Even on high-performance workstations, the execution time is in the order of days and memory requirements on the order of gigabytes, which impose restrictions on the type of systems that can be simulated. Parallel and distributed simulation (PDES) could be exploited to overcome these problems.

In this chapter, both sequential and parallel simulation tools for wireless mobile and ad hoc networks have been reviewed. We have also presented some recent examples of simulation methodologies to improve the simulation run time of these networks using PDES techniques.

ACKNOWLEDGMENTS

This work was supported by the Canada Research Program and Canada Foundation for Innovation (AB), and the Italian MIUR FIRB-PERF "Performance Evaluation of Complex Systems: Techniques, Methodologies and Tools" project funds (LB).

REFERENCES

1. R. Bagrodia, R. Meyer, et. al., "PARSEC: a Parallel Simulation Environment for Complex Systems," UCLA Technical report, 1997.

2. H. Bertoni, *Radio Propagation for Modern Wireless Systems,* Prentice-Hall, Upper Saddle River, NJ, 2000.

3. C. Bettstetter, “Smooth is Better than Sharp: a Random Mobility Model for Simulation of Wireless Networks,” in *Proceedings of ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM’01),* Rome, Italy, July 2001.

4. C. Bettstetter, H. Hartenstein, and X. Pérez-Costa, “Stochastic Properties of the Random Waypoint Mobility Model: Epoch Length, Direction Distribution and Cell-Change Rate,” in *Proceedings of the 5th ACM International Workshop,* MSWiM2002, September 2002.

5. C. Bettstetter, “Mobility Modeling in Wireless Networks: Categorization, Smooth Movement, and Border Effects,” *Mobile Computing and Communications Review, 5,* 3, July 2001.

6. L. Bononi, G. D’Angelo, and L. Donatiello “HLA-based Adaptive Distributed Simulation of Wireless Mobile Systems,” in *Proceedings of IEEE/ACM International Workshop on Parallel and Distributed Systems (PADS’03),* San Diego, CA, June 2003.

7. A. Boukerche, S. K. Das, and A. Fabbri “SWiMNet: A Scalable Parallel Simulation Testbed for Wireless and Mobile Networks,” *ACM/Kluwer Wireless Networks, 7,* 467–486, 2001.

8. A. Boukerche and A. Fabbri “Partitioning PCS Networks for Distributed Simulation,” in *IEEE High Performance Computing (HiPC),* LNCS 1970, Springer-Verlag, New York, pp. 449–458, 1970.

9. A. Boukerche, S. K. Das, A. Fabbri, and O. Yildiz, “Exploiting Model Independence for PCS Network Simulation,” in *Proceedings of ACM/IEEE Parallel and Distributed Simulation (PADS’99),* pp. 166–173, Atlanta, 1999.

10. A. Boukerche and C. Tropper,”Parallel Simulation on the Hypercube Multiprocessor,” in *Distributed Computing,* Spring Verlag, New York, 1993.

11. A. Boukerche, “Time Management in Parallel Simulation,” in *High Performance Cluster Computing,* Vol. 2, B. Rajkumar (Ed.), Prentice-Hall, Upper Saddle River, NJ, 1999.

12. J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva, “A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols,” in *Proceedings of MobiCOM’99,* Dallas Texas, October 1998; also at http://www.monarch.cs.cmu.edu

13. T. Camp, J. Boleng, and V. Davies, “A Survey of Mobility Models for Ad Hoc Network Research,” *Wireless Communications and Mobile Computing,* Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications, 2002.

14. C. Carothers, R. Fujimoto, Y.-B. Lin, and P. England, “Distributed Simulation of Large-scale PCS Networks,” in *Proceedings of the 2nd International Workshop on Modeling, Analysis, and Simulation of Computer and Telecommunication systems,* February 1994.

15. D. Cavin, Y. Sasson, and A. Schiper, “On the Accuracy of MANET Simulators,” in *Proceedings of POMC’02,* Toulouse, France, October 2002.

16. K. M. Chandy and J. Misra, “Distributed Simulation: A Case Study in Design and Verification of Distributed Programs,” *IEEE Transactions on Software Engineering, SE-5,* 440–452, September 1979.

17. S. R. Das and K. Jones, “Time-Parallel Algorithms for Simulation of MAC Protocols,” in *Proceedings of MASCOTS 2001,* Cincinnati, Ohio, August 2001.

18. S. Corson and J. Macker, “Mobile Ad Hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations,” RFC 2501, Jan. 1999.

19. DMSO: Defence Modeling and Simulation Office (1998), High Level Architecture RTI Interface Specification, Version 1.3, 1998.

20. FraSiMo, Framework for Simulation of Mobility in OMNET++, TKN Berlin, see http://www-tkn.ee.tu-berlin.de/research/research_texte/framework.html

21. R. M. Fujimoto, “Parallel Discrete Event Simulation,” *Communications of the ACM, 33,* 10, 30–53, October 1990.

22. R. M. Fujimoto, *Parallel and Distributed Simulation,* Wiley, New York, 2000.

23. Gerla M., Tang K., and Bagrodia R. "TCP Performance in Wireless Multi-hop Networks," in *Proceedings of IEEE WMCSA'99,* New Orleans, LA, February 1999.

24. A. C. Chandra, V. Gummalla, and J. O. Limb, "Wireless Medium Access Control Protocols," IEEE Communications Surveys and Tutorials, 2000.

25. Georgia Institute of Technology, RTI KIT, see http://www.cc.gatech.edu/computing/pads/fdk.html

26. J. Heidemann, N. Bulusu, J. Elson, C. Intanagonwiwat, K.-C. Lan, Y. Xu, W. Ye, D. Estrin, and R. Govindan, "Effects of Detail in Wireless Network Simulation," in *Proceedings of the SCS Multiconference on Distributed Simulation,* Phoenix, AZ, January 2001.

27. X. Hong, T. J. Kwon, M. Gerla, D. L. Gu, and G. Pei, "A Mobility Framework for Ad Hoc Wireless Networks," *Lecture Notes in Computer Science,* 2001.

28. X. Hong, M. Gerla, G. Pei, and C.-C. Chiang, "A Group Mobility Model for Ad Hoc Wireless Networks," in *Proceedings of ACM MSWiM'99,* Seattle, 1999.

29. R. Jain, *The Art of Computer Systems Performance Evaluation,* Wiley, New York, 1991.

30. D. R. Jefferson, "Virtual Time," *ACM Transactions on Programming Languages and Systems, 7,* 3 404–425, July 1985.

31. D. R. Jefferson and H. Sowizral, "Fast Concurrent Simulation Using the Time Warp Mechanism," in *SCS Multiconference on Distributed Simulation,* pp. 63–69, 1985.

32. P. Johansson, T. Larsson, N. Hedman, B. Mielczarek, and M. Degermark, "Scenario-based Performance Analysis of Routing Protocols for Mobile Ad Hoc Networks," in *Proceedings of MobiCOM'99,* pp. 195–206, Seattle, 1999.

33. K. Jones, "Parallel and Distributed Simulation Techniques and Applications," Ph.D. Proposal, November 2001, University of Cincinnatti.

34. T. S. Kim, J. K. Kwon, and D. K. Sung, "Mobility and Traffic Analysis in Three-Dimensional High-Rise Building Environments," *IEEE Transactions on Vehicular Technology, 49*, 5, 1633–1640, May 2000.

35. Y. Y. Kim and S. Q. Li, "Modeling Multipath Fading Channel Dynamics for Packet Data Performance Analysis," *Wireless Networks, 6,* December 2000.

36. C. Y. Lee William, *Mobile Cellular Telecommunications: Analog and Digital Systems,* McGraw-Hill, New York, 1989.

37. M. C. Little and D. L. McCue, "Construction and Use of a Simulation Package in C++," Computing Science Technical Report, University of Newcastle upon Tyne, Number 437, July 1993; also appeared in *C User's Journal, 12,* 3, March 1994.

38. M. Liljenstam and R. Ayani, "A Model for Parallel Simulation of Mobile Telecommunication Systems," in *Proceedings of the 4th International Workshop on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS),* San Jose, CA. 1996.

39. Y.-B. Lin and P. Fishwick, "Asynchronous Parallel Discrete Event Simulation," *IEEE Transactions on Systems and Cybernetics,* 1995.

40. M. Liljenstam, R. Ronngren, and R. Ayani, "Partitioning WCN Models for Parallel Simulation of Radio Ressource Management," *ACM/Kluwer Wireless Networks, 7,* 3, 307–324, 2001.

41. D. M. Lucantoni, M. F. Neuts, and A. R. Reibman, "Method for Performance Evaluation of VBR Video Traffic Models," *IEEE/ACM Transactions on Networking, 2,* 2, April 1994.

42. J. G. Markoulidakis, G. L. Lyberopoulos, D. F. Tsirkas, and E. D. Sykas, "Mobility Modeling in Third Generation Mobile Telecommunication Systems," *IEEE Personal Communications,* 41–56, August 1997.

43. R. A. Meyer and R. L. Bagrodia, "Improving Lookahead in Parallel Wireless Network Simulation," in *Proceedings of 6th International Workshop on Modeling, Analysis and*

Simulation of Computer and Telecommunication Systems, Montreal, Canada, pp. 262–267, July 1998.

44. MPI Primer, Developing with LAM, Ohio Supercomputer Center, The Ohio State University, 1996.
45. NS-2 Simulation Tool, see http://www.isi.edu/nsnam/ns/; NS-2 mobility extension from Rice Monarch, see http://www.monarch.cs.rice.edu/cmu-ns.html
46. A. Varga, OMNET++, in the column "Software Tools for Networking," *IEEE Network Interactive, 16,* 4, July 2002; also in http://whale.hit.bme.hu/omnetpp/
47. OPNET simulation tool, see http://www.mil3.com/home.html
48. J. Panchal, O. Kelly, J. Lai, N. Mandayam, A. Ogielski, and R. Yates, "Wippet, A Virtual Testbed for Parallel Simulations of Wireless Networks," in *PADS 98,* Banff, Canada, June 1998.
49. A. C. Palaniswamy and P. A. Wilsey, "An Analytical Comparison of Periodic Checkpointing and Incremental State Saving," in *Proceedings of the 1993 Workshop on Parallel and Distributed Simulation,* pp. 127–134.
50. Parallel and Distributed Network Simulator, PDNS, see http://www.cc.gatech.edu/computing/compass/pdns/
51. K. Perumalla, R. Fujimoto, and A. Ogielski, "A TED: A Language for Modeling Telecommunication Networks," *ACM SIGMETRICS Performance Evaluation Review, 25,* 4, March, 1998.
52. B. Ramamurthi, D. J. Goodman, and A. Saleh, "Perfect Capture for Local Radio Communications," *IEEE JSAC, SAC-5,* 5, June 1987.
53. T. S. Rappaport, *Wireless Communications: Principles and Practice,* 2nd ed., Prentice-Hall, Upper Saddle River, NJ, 2002.
54. E. M. Royer, P. M. Melliar-Smith, and L. E. Moser, "An Analysis of the Optimum Node Density for Ad Hoc Mobile Networks," in *Proceedings of IEEE International Conference on Communications (ICC),* Helsinki, June 2001.
55. S. Shah, E. Hernandez, and A. Helal, "CAD-HOC: A CAD Like Tool for Generating Mobility Benchmarks in Ad-Hoc Networks," in *Proceedings of SAINT'02,* Nara, Japan, February 2002.
56. W. Stallings, *Wireless Communications and Networks,* Prentice-Hall, Upper Saddle River, NJ, 2001.
57. Su, W. K. and C. L. Seitz, "Variants of the Chandy-Misra-Bryant Distributed Discrete Event Simulation Algorithm," in *Proceedings of the SCS Multiconference on Distributed Simulation,* Vol. 21, No. 2, 1989.
58. M. Takai, R. Bagrodia, K. Tang, and M. Gerla, "Efficient Wireless Network Simulations with Detailed Propagation Models," *ACM/Kluwer Wireless Networks, 7,* 3, 283–306, 2001.
59. M. Takai, J. Martin, and R. Bagrodia, "Effects of Wireless Physical Layer Modeling in Mobile Ad Hoc Networks," in *Proceedings of MobiHOC'01,* Long Beach, CA, October 2001.
60. J. Tian, J. Hahner, C. Becker, I. Stepanov, and K. Rothermel, "Graph-based Mobility Model for Mobile Ad Hoc Network Simulation," in *Proceedings of 35th Annual Simulation Symposium,* San Diego, CA, April 2002.
61. K. H. Wang and B. Li, "Group Mobility and Partition Prediction in Wireless Ad Hoc Networks," in *Proceedings of IEEE International Conference on Communications (ICC'02),* New York, April 2002.
62. A. Willig, "A New Class of Packet and Bit Level Models for Wireless Channels," in *Proceedings of 13th IEEE International Symposium on Personal Indoor and Mobile Radio Communications,* Lisbon, Portugal, 2002.
63. Wu, H. and R. Fujimoto, "Experiences Parallelizing a Commercial Network Simulator," in *Proceedings of the 2001 Winter Simulation Conference,* pp. 1353–1360.

64. J. Yoon, M. Liu, and B. Noble, “Random Waypoint Considered Harmful,” in *Proceedings of InfoCom,* 2003.

65. X. Zeng and R. Bagrodia, “GloMoSim: A Library for the Parallel Simulation of Large Wireless Networks,” in *Proceedings of the 12th Workshop on Parallel and Distributed Simulation,* Calgary, Canada, June 1998.

66. M. M. Zonoozi and P. Dassanayake, “User Mobility Modeling and Characterization of Mobility Patterns,” *IEEE Journal on Selected Areas in Communication, 15,* 7, September 1997.

CHAPTER 15

MODELING CROSS-LAYER INTERACTION USING INVERSE OPTIMIZATION

VIOLET R. SYROTIUK and AMARESH BIKKI

15.1 INTRODUCTION

Traditionally, networks are organized as a series of layers, each one built on the one below it. Although this simplifies the network design, the behavior of a protocol can vary significantly depending on the protocol above it or below it in the protocol stack. Many examples of cross-layer and inter-layer interaction are known. Although such interactions occur locally within a node in the network, they have also been found to occur between nodes that are not even adjacent [1]. Understanding how, and to what extent, protocols interact with each other will ultimately yield improvements in network performance.

The layered network design philosophy has largely predominated, even in wireless networks such as mobile ad hoc networks (MANETs). A MANET is a self-organizing collection of mobile wireless nodes with no supporting infrastructure. These networks are envisioned for situations such as emergency operations after a natural or environmental disaster has destroyed existing infrastructure, special operations in support of law enforcement activities, military missions in a hostile and/or unknown territory, commercial gatherings such as conferences, and the creation of interactive classrooms.

Recent research in MANETs has primarily focused on the optimization of protocols at individual layers of the protocol stack. Although there is an increasing awareness that protocols do not act in isolation, very little formal characterization of their interaction has been made. In Section 15.2, we overview the work related to protocol interaction in MANETs.

*Both authors were affiliated with the University of Texas at Dallas for the duration of this work.

Mobile Ad Hoc Networking. Edited by Basagni, Conti, Giordano, and Stojmenovic.
ISBN 0-471-37313-3 © 2004 Institute of Electrical and Electronics Engineers, Inc.

The problem motivating our work is the observation that routing protocols in MANETs using hop count as the path metric do not always route packets along the shortest hop path [2, 3]. For example, congestion in the network can cause the protocol to discover and route through paths other than the shortest hop path. Here, the congestion control mechanisms interact with the routing protocol to effectively change the link metrics into some more complex function than hop count. We have an inadequate understanding of the impact of congestion on the link metrics used in the shortest-path calculations. One way to improve our understanding of the link metrics used is to consider the inverse problem. Solving an inverse shortest-path problem consists of finding weights associated with the links of a network that are as close as possible to the a priori estimated values, and that are compatible with the observations of the shortest paths used for routing in the network.

In Section 15.3 we formulate an inverse shortest-path optimization problem to model the effect of congestion on routing in MANETs. We model congestion by the delay and the packet loss rate on each link in the network. Now, given actual routes computed by the routing protocol in the network, the question is: What are the link weights that make these routes the shortest paths? More specifically, we compute the contribution of each of the parameters of the delay and the packet loss rate to the link metric. The solution to the inverse shortest-path problem is obtained by formulating and solving a linear program. Section 15.4 describes our simulation study and demonstrates solutions that are compatible with observations in the simulated network. Finally, Section 15.5 summarizes the chapter.

15.2 RELATED WORK IN MANETs

First, we describe the interaction of the medium access control (MAC) layer with higher-layer protocols, and higher-layer interactions as observed in simulation studies. We then overview formal mechanisms to model protocol interaction.

15.2.1 MAC Protocol Interaction

Studies on the performance of MANET routing protocols run over different MAC protocols [4, 5, 6] conclude that table driven routing protocols are not affected by the selection of MAC protocol, whereas the amount of control traffic generated by a reactive routing protocol is dependent on the underlying MAC protocol. Furthermore, if a routing protocol generates more unicast packets than broadcast packets, this can affect throughput. This depends on the implementation of the MAC layer primitives. In some MAC protocols, such as IEEE 802.11 [7], there is more overhead in transmitting a unicast packet than a broadcast packet because a broadcast packet is not transmitted reliably.

In [8], MAC-layer mechanisms are isolated to determine the effect of each on network performance and to determine those most effective in supporting the User Datagram Protocol (UDP) and the Transport Control Protocol (TCP). The mechanisms considered include carrier sensing, packet sensing, carrier sensing with collision avoidance, handshake-control packets, and link-level acknowledgments (ACKs). The simulation study shows that carrier sensing alone is preferable for UDP traffic with a constant bit rate (CBR), whereas carrier sensing with collision avoidance is preferable for TCP running the file transfer protocol (FTP). In both cases, the handshake mechanism provides better sharing of the channel since it is a form of coordination among the nodes. Further, when ACKs are used, throughput tends to increase since unnecessary retransmissions are avoided.

Koksal et al. [9] examine how short-term unfairness of a MAC protocol can degrade the performance of transport and application protocols.

15.2.2 Routing Protocol Interaction

Several studies have examined the interaction of routing failures on TCP performance [3, 10, 11, 12]. In the Dynamic Source Routing (DSR) protocol [13, 14], routing failures can be the result of the propagation of stale routes from the cache. In TCP, if the source is not aware of a route failure it continues to transmit packets, leading to packet loss. Since packet loss is interpreted as congestion, TCP invokes congestion-recovery algorithms when the route is reestablished. This "slow start" mechanism throttles the transmission and results in performance degradation in the network.

There are several feedback-based schemes proposed in which the failure point notifies the source of route failures and/or reestablishments [12]. This allows a node to distinguish route failures from congestion and apply a different solution for each problem. Fixing retransmission timeout intervals has also been applied to distinguish between failure and congestion [15].

15.2.3 Formal Models for Protocol Interaction

To our knowledge, the first comprehensive study that attempts to characterize the interaction between the MAC and routing protocols in MANETs using more rigorous techniques is by Barrett et al. [1]. Extensive simulation-based experiments are performed and statistical analysis is used to study whether four factors R, v, M, and λ interact with each other in a significant way. Here R is a routing protocol, v is node velocity, M is a MAC protocol, and λ is a packet arrival rate. Statistically, interaction between two factors exists when the effect of a factor on a response variable is modified by another factor in a significant way. The response variables used are latency, number of packets received, and fairness.

Starting with a saturated model, backward elimination is applied. This method checks each k-way factor interaction term for significance using analysis of variance techniques, $k = 4, \ldots, 1$, and eliminates it if it is found to be insignificant. In this way, the smallest model that explains the simulation data is found.

The smallest model found when latency is the performance measure is the three-way $[R, v, M]$ term, whose interaction is due to the two-way interactions between $[R, M]$ and $[v, M]$. Interestingly, there is no interaction between $[R, v]$ on latency. For the measure number of packets received, all four factors $[R, v, M, \lambda]$ interact significantly, whereas for fairness the two-way interactions of $[R, M]$ and $[M, \lambda]$ are found to be most significant.

In [16], an approach based on source codes, combined with suitable routing algorithms and the reencoding of data at intermediate relays nodes, is used to capture the interdependencies between routing and data compression. This model is used to show that the amount of data generated by a sensor network is below its transport capacity. This supports the feasibility of large-scale multihop sensor networks.

15.3 CHARACTERIZING INTERACTION USING INVERSE OPTIMIZATION

We first describe an experiment that shows how congestion mechanisms impact routing. These observations motivate us to introduce a function of congestion in the link metric used for routing, and to use inverse optimization to find the weights associated with the links that are compatible with the actual observed shortest paths used for routing in the network.

15.3.1 Motivating Example

In an effort to understand the effect of congestion mechanisms on routing, we set up the static network topology shown in Figure 15.1 with each node running the Dynamic Source Routing (DSR; see Section 15.6 for a brief overview of the DSR routing protocol) [13, 14] protocol in the *ns-2* [17] network simulator. We use a static network to conclusively show that although DSR uses hop count as the path metric, the congestion mechanisms cause the protocol to discover and use a longer path even though a shorter path exists in the network. We use DSR for routing because it is proactive, and because of the convenience of the complete source–destination paths stored locally in the source's route cache.

Then we model a MANET by a weighted, undirected graph $G = (V, E, W)$. The vertex set V corresponds to the nodes in the network. An edge (i, j) exists in the edge set E whenever the vertices i and j are in the transmission range of each other. W is the set of edge weights, that is, $W = \{w_{i,j}\}$ for $(i, j) \in E$. A *path* $P_{s,d}$ of length k from a vertex s to a vertex d in a graph G is a sequence $\langle v_0, v_1, v_2, \ldots, v_k \rangle$ of vertices such that $s = v_0$, $d = v_k$, and $(v_{i-1}, v_\text{i}) \in E$ for $i = 1, 2, \ldots, k$. The length of the path $P_{s,d}$, denoted as len($P_{s,d}$), is the number of edges in the path.

We induced a large volume of traffic between nodes 3 and 4 and, as a result, the link between nodes 3 and 4 is heavily congested. Now consider what happens when node 1 initiates a route request (RREQ) packet to the destination node 7 once link (3, 4) is congested. There are two paths through which the RREQ packet can reach the destination.

The one path of length six is $P_{1,7} = \langle 1, 2, 3, 4, 5, 6, 7 \rangle$ and the other path of length seven is $P'_{1,7} = \langle 1, 2, 8, 9, 10, 11, 6, 7 \rangle$. Since the link between nodes 3 and 4 is congested, the route request may be dropped at node 3. Some reasons the RREQ packet may be dropped include: buffer overflow at node 3, collision of the packet with data traffic in the opposite direction, i.e., from node 4 to 3, and queueing delays at node 3.

If the route request is dropped at node 3, then node 7 may only receive one RREQ that has traversed the longer path $P'_{1,7}$. As a result, node 1 receives a route reply (RREP) from node 7 containing route $P'_{1,7}$. This route is then cached and used by node 1 to send traffic to node 7.

If the route request is delayed at node 3 rather than dropped, then the RREQ along path $P'_{1,7}$ reaches node 6 earlier than from along $P_{1,7}$. In this case, node 6 discards the second RREQ as it has the same request identifier as that of the RREQ from the longer, but less congested, path. Therefore, even in the event of a delay along the shorter path, node 1 receives a RREP that contains the route $P'_{1,7}$.

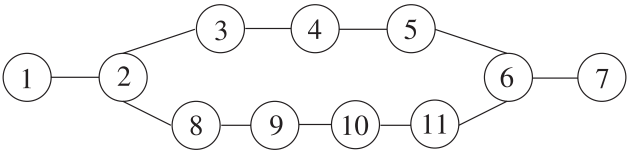


Figure 15.1. Static MANET topology demonstrating effects of congestion.

Suppose that this time node 1 performs route discovery to node 6. Suppose that, as a result, it found two routes to node 6, $P_{1,6} = \langle 1, 2, 3, 4, 5, 6\rangle$ of length five, and $P'_{1,6} = \langle 1, 2, 8, 9, 10, 11, 6\rangle$ of length six. Since the first route is shorter than the second, node 1 sends any packets destined to node 6 through path $P_{1,6}$ (now, duplicate RREQs do not occur at intermediate nodes).

Consider the scenario in which node 1 is routing packets to node 6 and link (3, 4) becomes congested. Now, a packet is dropped at node 2 because it exceeded the number of retransmissions when node 2 tried to forward it to node 3. This might happen if node 3 is too busy forwarding its own traffic to node 4. In such a case, node 2 sends a route error (RERR) packet back to node 1, informing it that the link between nodes 2 and 3 is no longer available. Upon reception of this packet, node 1 purges the route $P_{1,6}$ from the cache and starts using the route $P'_{1,6}$, which is already present in its route cache, to send a packet to node 6. Despite the existence of a shorter path, node 1 starts to use a longer path.

This example demonstrates that the presence of congestion may result in a node routing packets over a path other than the shortest-hop count path. Hence, congestion effectively alters the link metrics in the network. This suggests that rather than $w_{i,j} = 1$ for all $(i, j) \in E$, its value should reflect the congestion on the links.

15.3.2 Modeling Congestion in the Link Metric

In order to reflect the effect of congestion on routing, we model the link weight as a linear function of the hop count, the average packet delay ($d_{i,j}$), and the average packet loss rate ($\ell_{i,j}$) across a link (i, j), that is,

$$w_{i,j} = 1 + \alpha \cdot d_{i,j} + \beta \cdot \ell_{i,j} \tag{15.1}$$

where the values of α, $\beta > 0$ represent the relative weights of the packet delay and loss rate, respectively, in the cost of a link. Although, in reality, link weight may be a more complex function of congestion (and other parameters) than we use in our model, we choose these two parameters to validate our idea.

Now, the cost of a path $P_{s,d}$ between a source node s and a destination node d ($w_{P_{s,d}}$) is defined as follows:

$$w_{P_{s,d}} = \sum_{(i,j)\in P_{s,d}} w_{i,j} = \text{len}(P_{s,d}) + \alpha \sum_{(i,j)\in P_{s,d}} d_{i,j} + \beta \sum_{(i,j)\in P_{s,d}} \ell_{i,j} \tag{15.2}$$

We are interested in finding the values of α and β in Equation (15.2) to determine how, and to what extent, these elements of congestion affect the routing protocol. In order to accomplish this, we formulate an inverse shortest-path problem.

15.3.3 Inverse Shortest-Path (ISP) Formulation

When solving an optimization problem, usually the parameters such as costs, capacities, and so on are known and the interest is in finding an optimal solution. However, in practice, certain solutions might be known to be optimal from observations or experiments. The idea of inverse optimization is to find values of the parameters that make the known solutions optimum and that differ from the estimates as little as possible. Heuberger and

Ahuja et al. provide a good general introduction to inverse optimization [18, 19], with interesting applications in geophysics, medical imaging, transportation, and communication.

An important "forward" problem in graph theory (networks) is that of finding shortest paths (routes) between one or more pairs of vertices (nodes) in a graph. A shortest path is a path that is better than all other paths available between the same source and destination pairs. A path is considered better if it is optimized on some predetermined weight, such as the sum of costs of edges that form the path. There are various algorithms to solve the shortest-path problem, such as those proposed by Dijkstra [20], Bellman [21], Ford [22], and Moore [23].

In the *inverse shortest-path* (ISP) problem, we are given a weighted graph and a set of paths. The task is to modify the edge weights as little as possible such that the given paths become optimal paths between the corresponding source and destination.

For example, consider the graph in Figure 15.2 with the edge weights equal to one, and the path $P_{1,7} = \langle 1, 2, 3, 4, 6, 7 \rangle$ between source node 1 and destination node 7 as optimal. If all edge weights remain one, $P_{1,7}$ cannot be an optimal path, as a path $\langle 1, 2, 5, 6, 7 \rangle$ of length four exists. In order to make $P_{1,7}$ optimal, we must modify edge weights. It can be easily seen that by setting the weights of edges of (2,5) and (5, 6) each to 1.5, the given path becomes optimal. (Note that the solution is not unique in this case.)

More formally, the inverse shortest-path problem takes, as input, a weighted, undirected graph $G = (V, E, W)$ and a set $\mathrm{P} = \{P_{s,d}\}$ of paths in the graph between certain (but not necessarily all) pairs s and d of nodes. The task is to find w', the modified weights of the edges, such that

- $\min \| w' - w \|$
- Each given path in P arises as a minimum weight path between its endpoints.

We consider the use of the ℓ_1 norm, that is, the absolute value, in the objective function.

The inverse shortest-path problem was first studied in detail by Burton [24]. He proposed various methods to solve the inverse shortest-path problem. One method introduces the concept of an island to characterize the violation of the shortest-path constraint by the given optimal paths. Then it uses the method of Goldfarb et al. [25] to solve the resulting quadratic programming formulation.

We use an approach similar to the one presented in [26] to solve our inverse shortest-path problem. This approach formulates the problem as a linear program. We then use the interactive optimizer software CPLEX [27] to solve the resulting linear programming problem. Since the inverse shortest-path problem that occurs in our work has the characteristic that $w' \geq w$, we can ignore the absolute value function in the objective function.

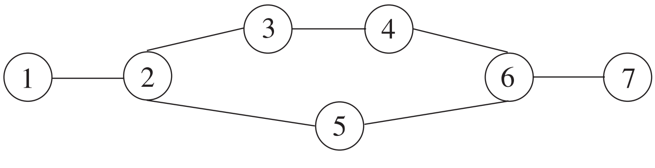


Figure 15.2. Instance of inverse shortest-path problem.

The linear programming [28] formulation of our inverse shortest paths problem is:

Objective function:

$$\min\left\{\sum_{(i,j)\in E}(w'_{i,j}-w_{i,j})\right\} \tag{15.3}$$

Constraints:

$$S_{i,j}\geq 0 \qquad \forall i,j \tag{15.4}$$

$$w'_{i,j}\geq w_{i,j} \qquad \forall (i,j)\in E \tag{15.5}$$

$$S_{i,k}+w'_{k,j}\geq S_{i,j} \qquad \forall i,j,k;\,(k,j)\in E \tag{15.6}$$

$$\sum_{(u,u')\in P_{i,j}} w'_{u,u'}\leq S_{i,j} \qquad \forall i,j;\, P_{i,j}\in \mathcal{P} \tag{15.7}$$

Since the task is to modify the edge weights as little as possible, we need to minimize the difference between the old edge weights ($w_{i,j}$) and the new edge weights ($w'_{i,j}$). This is indicated by the objective function [Equation (15.3)]. All the weights of edges are positive resulting in shortest paths ($S_{i,j}$) of positive length. This constraint is reflected by Equation (15.4). In our work, the modified edge weights are always greater than the original values, as shown in Equation (15.5). The cost of the shortest path between a source *s* and a destination *d* is less than or equal to the sum of the cost of the shortest path between *s*, a neighbor of *d*, and the cost of the edge between this neighbor and *d*. This principle of path optimality is captured in Equation (15.6). The Equation (15.7) reflects the fact that the given optimal paths have a cost less than the shortest path between the same vertices.

Note that we are not directly interested in computing the value of the modified link weights. We are instead interested in computing the values of α and β in the modified link weights. From Equation (15.1), we see that the modified link weights are a linear function of α and β. Note that the delay and the loss rate parameters of this equation are *not* variables. Their values are computed through simulation. By substituting the modified link weights ($w'_{i,j}$) as defined in Equation (15.1), we obtain a new linear programming formulation that has α and β as the variables:

Objective function:

$$\begin{aligned}&\min\left\{\sum_{(i,j)\in E}(w'_{i,j}-w_{i,j})\right\}\\&=\min\left\{\sum_{(i,j)\in E}(1+\alpha\cdot d_{i,j}+\beta\cdot \ell_{i,j})-1\right\}\\&=\sum_{(i,j)\in E}(\alpha\cdot d_{i,j}+\beta\cdot \ell_{i,j})\end{aligned} \tag{15.8}$$

Constraints:

$$S_{i,j}\geq 1 \qquad \forall i,j \tag{15.9}$$

$$\alpha\geq 0 \tag{15.10}$$

$$\beta \geq 0 \tag{15.11}$$

$$S_{i,k} + 1 + \alpha \cdot d_{i,j} + \beta \cdot \ell_{i,j} \geq S_{i,j} \qquad \forall i, j, k; (k, j) \in E \tag{15.12}$$

$$\sum_{(u,u')\in P} (1 + \alpha \cdot d_{u,u'} + \beta \cdot \ell_{u,u'}) \leq S_{i,j} \tag{15.13}$$

We now describe our simulation study and demonstrate that the solutions obtained for this linear programming formulation are compatible with observations in the simulated network.

15.4 SIMULATION STUDY

We describe how we set up the experiments in our simulation study by describing in detail one scenario. Then, we show the results of the study and interpret the resulting values of α and β for the link metrics.

15.4.1 Simulation Environment

For this work, we use *ns-2* (version ns-2.1b8) [17], a discrete-event simulator developed at Lawrence Berkeley National Laboratory. The wireless extensions added to *ns-2,* which include support for MANETs, were implemented by the Monarch Project at CMU [29].

In order to isolate the effect of congestion mechanisms on routing, we consider ten manually generated static network topologies. Each network contains 30 nodes in a 500 × 500 m^2 area. Each node in the network has omnidirectional transmission range of 100 m.

Two types of traffic are introduced into the network. One type is established between a source and destination and is used to observe the effects of congestion on routing, and the other type is used to create the congestion itself. Congestion is introduced by strategically injecting traffic between node pairs with the aim of creating an interference with potential routes taken by traffic between a source and destination. The traffic between a source and destination is less aggressive so that it does not create congestion in the network by itself.

For our simulations, TCP traffic is used between a source and destination. To create different levels of congestion in the links of the network, we use ten traffic patterns (including both UDP with CBR, and TCP traffic) for the traffic between the other nodes. In total, we run 10 × 2 × 10 = 200 simulations, with ten static topologies, two kinds of traffic between a source and destination, and ten traffic patterns for the other nodes.

Each simulation is run for 50 seconds of real time. The traffic in the simulation starts randomly between 0.5 s and 1.0 s and continues until the end of the simulation. All of the nodes in the network are configured with a drop-tail priority interface queue with a buffer length of 50, DSR as the routing protocol, and IEEE 802.11 as the MAC protocol.

The source in the network traces its route cache periodically with a time interval of 0.1 s. The MAC trace and routing trace are enabled on all nodes to enable calculation of the average packet delay and loss rate across each link.

15.4.2 Example Scenario

One of the ten static topologies used is a 5 × 6 grid of nodes, as shown in Figure 15.3. The source and destination nodes are nodes 1 and 22, respectively. The traffic flowing between them is continuous TCP/Reno traffic with a window size of 32 and a packet size of 64

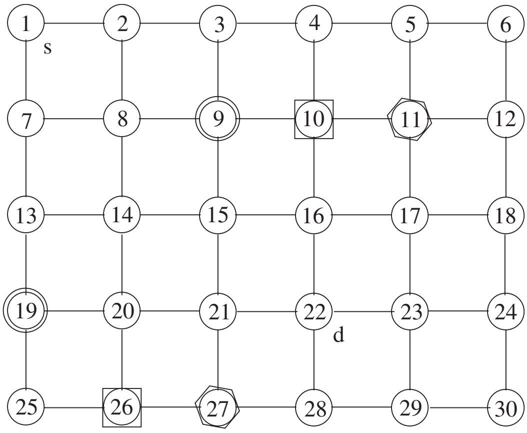


Figure 15.3. Static 5 × 6 grid network topology. *s* and *d* denote the source and destination nodes. Traffic between the three indicated pairs of nodes is used to induce congestion.

bytes. To induce congestion in the network, CBR traffic over UDP is introduced between node pairs 9 and 19, 10 and 26, and 11 and 27, with a packet arrival interval time of 0.006 s, 0.006 s, and 0.1 s, respectively, to create dynamic levels of congestion. The packet size of all of the CBR traffic is 64 bytes.

15.4.3 The Route Cache Trace

The route cache trace file at the source node contains entries consisting of a timestamp followed by routes from the primary and secondary route cache at that time. In DSR, the primary route cache contains the routes found using the route-discovery mechanism. The secondary route cache contains the routes learned from promiscuous listening. When a packet is sent to a destination, both the primary and the secondary cache entries are searched for the minimum hop-count path. If the route found is from the secondary cache, it is promoted to the primary cache.

The route cache trace file is processed to produce a sequence of the paths observed to route packets for the duration of the simulation. Figure 15.4 shows the length of each path in the sequence for the duration of the simulation. In this run, the length of the shortest path in the cache varies between six and eight. From time to time, the cache is empty, indicated by a path of length zero. This may happen when a route error purges routes in the cache due to a link failure, or when the cache entries time out.

The figure shows that, for most of the time in the simulation, the traffic between the source and destination follows a path of length eight. This corresponds to the route ⟨1, 2, 3, 4, 5, 11, 17, 23, 22⟩. The paths of length six intersect the links on which congestion is being induced, hence DSR avoids using the shorter paths and instead uses the longer path having lower average packet delay and loss rates.

15.4.4 Computing the Packet Delay and Loss Rate

In order to compute the delay and loss rate for each link in the network, the MAC and routing trace entries are analyzed. Data packets (i.e., TCP, CBR), ARP packets, and DSR

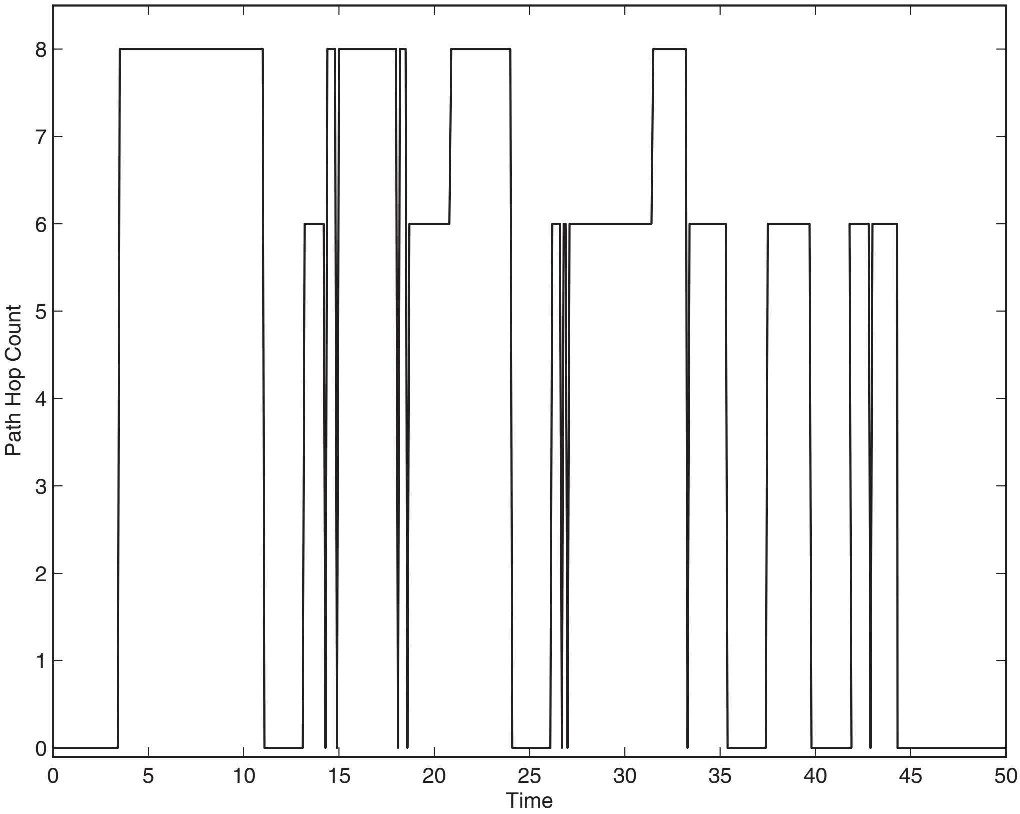


Figure 15.4. Length of routes in a route cache trace.

packets (i.e., RREQ, RREP, RERR) are all taken into account to compute the average delay and loss rate (percentage of packets dropped). See Bikki [2] for detailed trace files.

15.4.5 Solving the Linear Programming Formulation

CPLEX is a software product from ILOG, Inc. [27] that solves linear, mixed-integer, and quadratic programming problems. We use this software to solve the linear programming formulation of our ISP problem. For each nonzero time interval in the route cache trace, the route used, the average packet delay, and the average packet loss rate for all the links are used as input for the linear program. This program is fed into CPLEX and it produces an output file that contains values of α and β for that time interval. To obtain the value for α and β for a simulation run, we average the values over the nonzero time intervals in the route cache trace.

Figures 15.5 and 15.6 show the values of α and β, respectively, for each of the 200 simulation runs of our experiment.

15.4.6 Interpretation of α and β

It is evident from Figures 15.5 and 15.6 that there are three distinct regions of values for the coefficients. The first region (simulation runs 1 to 40) corresponds to simulations in

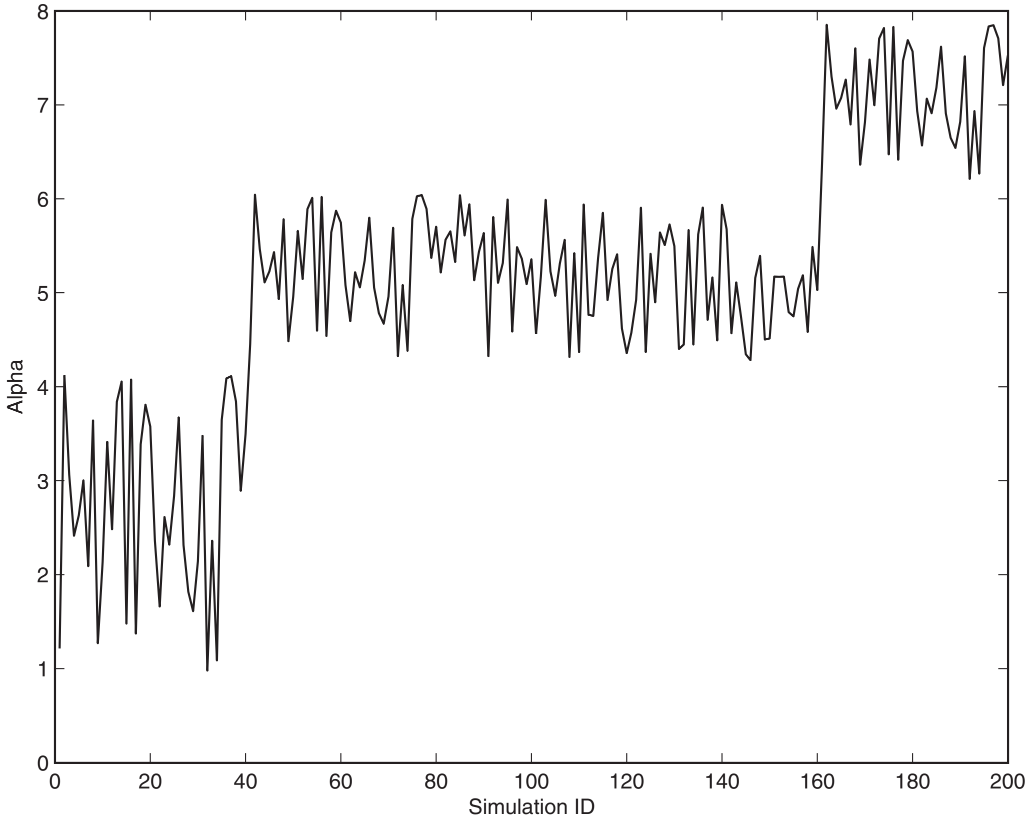


Figure 15.5. Delay coefficients (α).

which the two shortest routes between the observed source and destination have the same length in hop count. The second region (runs 41 to 100) corresponds to simulations with a one hop count difference between the two shortest routes. Finally, the third region (runs 161 to 200) corresponds to simulations with a two hop count difference between the two shortest routes. (Note that there may also be other, higher hop count routes present between the source and destination.)

For the first region, there are some time periods in the route cache trace in which the difference in the length of the observed path and the next-best path is equal to zero. In this case, any small values for α and β satisfy the requirement of Equation (15.1). These values of α and β keep the overall average α and β values for the entire simulation low. For the simulations in the second region, route trace contains time periods in which the difference in the length of the observed path and the next-best path is equal to one. Now, in order to satisfy the requirement of Equation (15.1), the inverse shortest-path problem needs to assign larger values for α and β, and similarly for the third region of simulations.

Thus, for increasing distance between the shortest hop count path between a source and destination and the path actually used by the routing protocol, we see increasing coefficients on the congestion parameters in the link metric.

One way that the α and β values may be used is as follows. Network designers can compute the values of average packet delay and average packet loss rate by the use of targeted

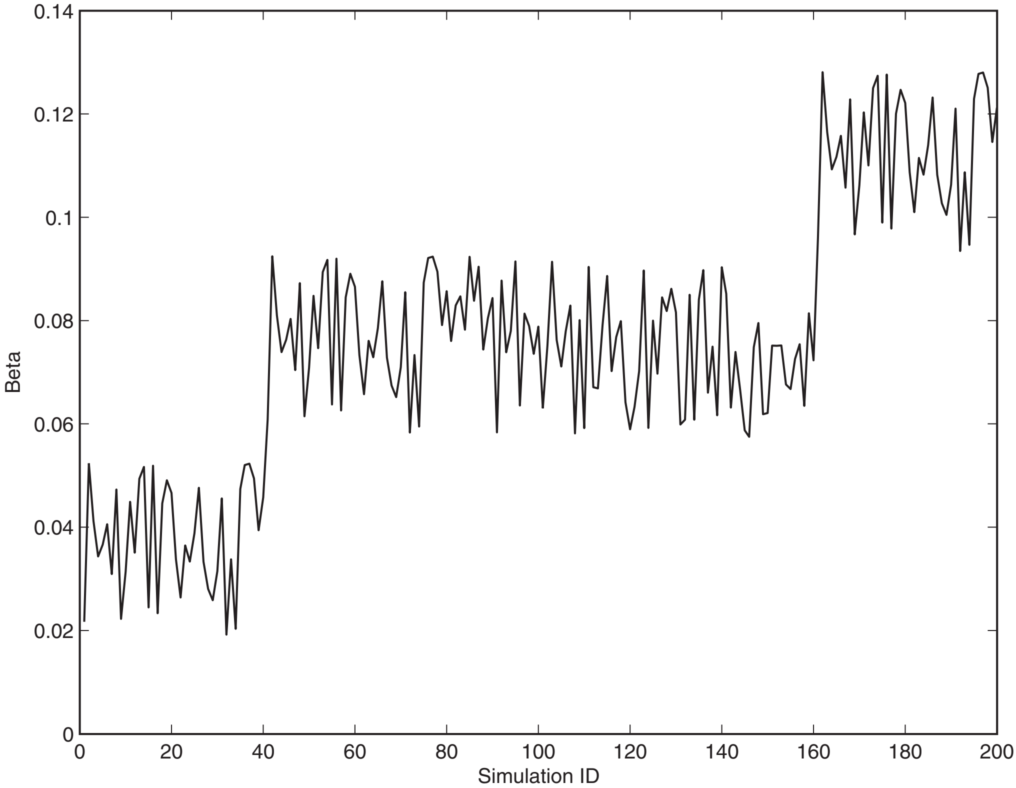


Figure 15.6. Packet loss coefficients (β).

traffic patterns and node configurations. Using the equations in Section 15.3.3, we can obtain threshold values for the delay and loss rate below which DSR uses the weight solely of hop count when routing the traffic. When the delay and loss rates in the current path exceed the threshold values, DSR tries to select a longer path with low delay and loss rates.

In certain task-specific MANETs, it may be desirable to have the traffic take a minimum hop count path to minimize the overall traffic delay. In such cases, the above equation can be used to obtain the maximum fluctuations of the delay and loss rates that would still satisfy the desired constraint.

15.5 THE DYNAMIC SOURCE ROUTING (DSR) PROTOCOL

The Dynamic Source Routing (DSR) protocol [13, 14] is a reactive protocol that uses *source routing* to route traffic in the network. Source routing is a mechanism in which the source of a packet determines the complete sequence of nodes the packet needs to follow in order to reach the destination. This sequence of nodes is placed in the packet header, so that all the intermediate nodes can determine the next hop to which it should forward the packet directly.

In DSR, each mobile node in the MANET maintains a route cache. This cache is used to store the source routes learned by the node. When a source node wants to send a packet to a destination, the source checks its cache to see if it has a source route to that destination. If a route is found, then the packet is transmitted after inserting the source route into the packet header. If no route is found for that destination, then the source initiates the route discovery mechanism. The packet is buffered while the route discovery mechanism is underway.

A host initiates the route discovery by broadcasting a route request (RREQ) packet. This packet includes information such as the initiator of this RREQ, the destination that is the target of this RREQ, and a route sequence list that contains the list of node identifiers traversed by the route request. It also contains a request identifier (id), set by the initiator. This is a sequence number that is unique to each route request initiated at this node. Initially, the route sequence list is set to the source id at the initiator of the route discovery.

Each node in the network maintains a list of initiator id, request id pairs from the most recent route request. All intermediate nodes that receive a RREQ forward it if: (1) the pair ⟨initiator id, request id⟩ from the RREQ packet does not match any entry from the list it is maintaining, and (2) its node id is not already present in the route sequence list of the RREQ.

The rebroadcast explosion of duplicate route requests is suppressed by discarding a route request if the node has recently seen another route request belonging to the same route discovery. By not forwarding a route request when the node id is already present in the node list, DSR ensures that no loop is present in the route realized by the route discovery mechanism.

When the RREQ reaches the target destination, the route reply (RREP) is sent back to the initiator of the route discovery. The RREP contains the route sequence list from the corresponding RREQ appended with the destination node id.

The route maintenance mechanism performs the job of validating the routes. While the route is in use, this mechanism monitors the operation of the route and informs the sender of any routing errors by sending a route error (RERR) packet to the original sender of the packet. The route error packet contains the ids of the endpoints of the link that is no longer available. Upon receiving a route error, a node purges the route entries with this link from its route cache.

The packets in the cache and in the buffer are periodically validated. A packet is purged from the buffer when it has been buffered for longer than the expiration period. Similarly, each entry in the route cache has an expiration time period after which it is removed from the cache.

For further details and optimizations of DSR, the interested reader should consult references [13, 14].

15.6 CONCLUSIONS

In this chapter, we have proposed the use of the inverse shortest paths problem as a method to characterize the effect of congestion on routing protocols in MANETs, and demonstrated solutions of the resulting linear programming problem that are compatible with observations in the simulated network. In general, inverse optimization may provide a promising formal framework for understanding cross-layer and inter-layer protocol interaction.

ACKNOWLEDGMENT

We are grateful to Professor A. Faragó for helpful discussions as this work evolved.

REFERENCES

1. C. Barrett, M. Drozda, A. Marathe, and M. V. Marathe, "Characterizing the Interaction Between Routing and MAC Protocols in Ad Hoc Networks," in *Proceedings of the Third ACM International Symposium on Mobile Ad Hoc Networking and Computing* (MobiHoc'02), pp. 92–103, Lausanne, Switzerland, 2002.
2. A. Bikki, "Using Inverse Optimization to Model the Effect of Congestion on Routing Protocols in MANETs," M. S. Thesis, University of Texas at Dallas, August 2002.
3. G. Holland and N. Vaidya, "Analysis of TCP Performance over Mobile Ad Hoc Networks," in *Proceedings of the Fifth ACM Conference on Mobile Networking and Computing* (MobiCom'99), pp. 219–230, Seattle, 1999.
4. E. M. Royer, S.-J. Lee, and C. E. Perkins, "The Effects of MAC Protocols on Ad hoc Network Communication," in *Proceedings of the IEEE Wireless Communications and Networking Conference* (WCNC'00), Volume 2, pp. 543–548, Chicago, 2000.
5. S. R. Das, C. E. Perkins, and E. M. Royer, "Performance Comparison of Two On-Demand Routing Protocols for Ad Hoc Networks," in *Proceedings of the Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies* (Infocom'00), pp. 3–12, Tel Aviv, Israel, 2000.
6. C. E. Perkins, E. M. Royer, S. R. Das, and M. K. Marina, "Performance Comparison of Two On-Demand Routing Protocols for Ad Hoc Networks," *IEEE Personal Communications Systems (PCS) Magazine,* special issue on Mobile Ad Hoc Networks, *8,* 1, 16–29, February 2001.
7. *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications,* IEEE 802. 11 Standard, IEEE, New York, 1996.
8. K. Tang, M. Correa, and M. Gerla, "Effects of Ad Hoc MAC Layer Medium Access Mechanisms Under TCP," *ACM/Kluwer Mobile Networks and Applications* (MONET), *6,* 4, 317–329, August 2001.
9. C. E. Koksal, H. Kassab, and H. Balakrishnan, "An Analysis of Short-Term Fairness in Wireless Media Access Protocols," in *Proceedings of the ACM International Conference on Measurement and Modeling of Computer Systems* (SIGMETRICS'00), 118–119, Santa Clara, California, 2000.
10. A. Ahuja, S. Agarwal, J. P. Singh, and R. Shorey, "Performance of TCP over Different Routing Protocols in Mobile Ad-Hoc Networks," in *Proceedings of the IEEE Vehicular Technology Conference* (VTC'00), Vol. 3, pp. 2315–2319, Tokyo, Japan, 2000.
11. G. Holland and N. Vaidya, "Impact of Routing and Link Layers on TCP Performance in Mobile Ad Hoc Networks," in *Proceedings of the IEEE Wireless Communications and Networking Conference* (WCNC'99), Vol. 3, pp. 1323–1327, New Orleans, Louisiana, 1999.
12. K. Chandran, S. Raghunathan, S. Venkatesan, and R. Prakash, "A Feedback-Based Scheme for Improving TCP Performance in Ad Hoc Networks," *IEEE Personal Communications Systems (PCS) Magazine,* special issue on Ad Hoc Networks, *8,* 1, 34–39, February 2001.

13. D. Johnson and D. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks," in T. Imielinski and H. Korth (Eds.), *Mobile Computing,* pp. 153–181, Kluwer Academic Publishers, 1996.

14. D. Johnson, D. Maltz, Y. Hu, and J. Jetcheva, "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks," IETF Internet Draft, work in progress. http://www.ietf.org/internet-drafts/draft-ietf-manet-dsr-07.txt

15. T. D. Dyer and R. V. Boppana, "Comparison of TCP Performance over Three Routing Protocols for Mobile Ad Hoc Networks," in *Proceedings of the 2001 ACM International Symposium on Mobile Ad Hoc Networking and Computing* (MobiHoc'01), pp. 56–64, Long Beach, California, 2001.

16. A. Scaglione and S. Servetto, "On the Interdependence of Routing and Data Compression in Multi-Hop Sensor Networks," in *Proceedings of the Eighth ACM International Conference on Mobile Computing and Networking* (MobiCom'02), pp. 140–147, Atlanta, Georgia, 2002.

17. Network Simulator, ns-2. The VINT Project. http://www.isi.edu/nsnam/ns/

18. C. Heuberger, "Inverse Combinatorial Optimization: A Survey on Problems, Methods and Results," to appear in *Journal of Combinatorial Optimization*.

19. R. K. Ahuja and J. B. Orlin, "Inverse Optimization," *Operations Research, 49,* 771–783, 2001.

20. E. W. Dijkstra, "A Note on Two Problems in Connexion with Graphs," *Numerische Mathematick, 1,* 269–271, 1959.

21. R. Bellman, "On a Routing Problem," *Quart. Appl. Math., 16,* 87–90, 1958.

22. R. W. Ford, "Algorithm 97: Shortest Path," *Communications of the ACM, 5,* 345, 1962.

23. E. F. Moore, "Shortest Path through a Maze," in *Proceedings of the International Symposium on the Theory of Switching,* Part III, Harvard University, Cambridge, MA, pp. 285–292, 1959.

24. D. Burton, "On the Inverse Shortest Path Problem," Doctoral Dissertation, Facultés Universitaires Notre-Dame de la Paix de Namur, Faculté des Sciences, Départment de Mathématique, 1993.

25. D. Goldfarb and A. Idnani, "A Numerically Stable Dual Method for Solving Strictly Convex Quadratic Programs," *Mathematical Programming, 27,* 1–33, 1983.

26. A. Faragó A. Szentesi, and B. Szviatovszki, "Inverse Optimization in High Speed Networks," to appear in *Discrete Applied Mathematics,* special issue on Combinatorial and Algorithmic Aspects of Telecommunications.

27. CPLEX Interactive Optimizer, ILOG, Inc. http://www.ilog.com

28. V. Chvátal, *Linear Programming,* W.H. Freeman and Company, New York, 1983.

29. Wireless and Mobility Extensions to *ns-2*. Carnegie Mellon University, Monarch (Mobile Networking Architectures) Project. http://www.monarch.cs.cmu.edu

CHAPTER 16

ALGORITHMIC CHALLENGES IN AD HOC NETWORKS

ANDRÁS FARAGÓ

In this chapter, we review a number of algorithmic problems, primarily motivated by ad hoc networks, especially by routing protocols in these networks. We show that the amazing diversity of possible route metrics, each with its own justifiable motivation, can be incorporated into a unified mathematical framework. A number of unsolved algorithmic problems are presented in connection with route metrics, as well as other path-related problems, such as mobile paths in mobile graphs, the inverse shortest-path problem and finding large route systems. Overall, we would like to convince the reader that ad hoc networking provides many interesting challenges not only for practical implementation, but also for the more theoretical side of algorithm development and analysis.

16.1 INTRODUCTION

Ad hoc networks present a rich set of new challenges for algorithm development. Our goal in this chapter is to review a selected set of such problems, showing that ad hoc networking (and, for that matter, networking in general) can contribute a fertilizing effect to the research on algorithms.

Since it would be impossible to provide an exhausitve description of all ad hoc network related algorithm design issues in a single chapter, we used the following principles in the selection and presentation:

- We focus on ad hoc networking motivated issues that can be formalized as algorithmic problems on graphs. Since this is still more than what can be addressed in a

Supported in part by NSF Grants ANI-0105985 and ANI-0220001.

Mobile Ad Hoc Networking. Edited by Basagni, Conti, Giordano, and Stojmenovic.
ISBN 0-471-37313-3 © 2004 Institute of Electrical and Electronics Engineers, Inc.

chapter, we specifically selected the area of *path problems* in graphs. The reason for focusing on graphs is that they are perhaps the best-known structures in the field of discrete algorithms and at least the questions can be understood without too much background. The reason for selecting path problems is that this is related to routing in networks, which is a very rich area of algorithmic issues.

- Since we are interested in this chapter in core algorithmic issues, rather than practical implementations, therefore, we aim at exhibiting the essence of each question in a mathematically meaningful way. Even though the ultimate motivation is practical ad hoc networking, in the models we "peel off" various layers of practical issues that are piled on the core algorithmic problem. This approach sometimes allows us to generalize the question, following its own intrinsic logics, yielding some interesting unsolved problems.
- Our main interest is in those problems that offer some chance for an efficient solution, at least under some restrictions, but, at the same time, they are not fully solved yet. That is, we try to walk in the narrow region that separates the "plane" of solved problems from the "sky high muountains" of hopelessly difficult tasks.

The reader is assumed to be familiar with the basic concepts of graph theory and algorithms, including *NP*-completeness.[1]

16.2 A FRESH LOOK AT SHORTEST PATHS

An amazing variety of ad hoc routing protocols exists (see e.g., [16, 19, 21] and references therein). No matter how different they may be, however, in every routing protocol it is a key common task to find a "good" path between a source and a destination node.

But which path is good enough? If we have a path metric (such as hop count, expected delay, expected lifetime, etc.), then, naturally, it is desirable to find a path that is optimal or at least nearly optimal with respect to the given path metric. Nevertheless, this core algorithmic task is often overshadowed by a multitude of other aspects. When addressing routing protocols, we usually talk about a number of features, such as how the protocol can be proactive or reactive, source-initiated or table-driven, link state or distance vector, flat or hierarchical, and so on. This is not surprising, since in an ad hoc routing protocol one typically has to find/maintain/update routes in a distributed way. In any case, however, we still have the common, fundamental graph problem of finding a path according to some route metric that is either given explicitely or defined implicitely by the protocol details. This fundamental task of path finding is the one at which we take a fresh look in this section.

One may immediately ask: does the finding of a (static) shortest path present any serious algorithmic challenge? After all, it is among the most basic algorithmic tasks in graph theory and was very well solved in the 1950s.[2] Well, our answer is: it all depends on the

[1] We use the usual distinction between *NP*-complete and *NP*-hard, since most of the optimization tasks ask for more than a yes/no answer and, therefore, they cannot fall directly in the complexity class *NP*. A task Q is *NP*-hard if an *NP*-complete problem can be polynomially reduced to it (and, therefore, all *NP*-complete problems can be polynomially reduced to it), but Q itself is not necessarily in the class *NP*.

[2] Dijkstra's algorithm appeared in 1959 [9]. Bellman published the precursor of the Bellman–Ford algorithm in 1958 [4].

path metric. Let us exhibit below a few examples of path metrics among which quite a few apparently give rise to harder tasks than the conventional shortest-path problem, yet they still have a natural motivation. In each example, we search for a simple path between two given nodes u and v (simple means that no repeated edges or nodes are allowed on the path; this will be required everywhere throughout this chapter). A path between nodes u and v is called a u–v path, but we often just call it a path if no ambiguity arises. A path is regarded as a set of links (edges in the graph theoretic terminology).

16.2.1 Examples of Path Problems

Most reliable delay constrained path. For each link, a delay value and a reliability value are known. The end-to-end delay is the sum of the link delays along the path, whereas the path reliability is the product of the link reliabilities. Find the most reliable path among those that obey a given end-to-end delay bound.

Least vulnerable path. Certain sets $E_1, E_2, \ldots, E_k$ of links are specified that are threatened by some attack or jamming. The links in any given set E_i are likely to fail together. Find a path that intersects with the smallest number of the given link sets.

Minimum exposure path. Given a subset of the nodes that are in danger of attack or jamming, find a path that is the farthest from these nodes, that is, the smallest occuring hop distance between a path node and an endangered node is maximum. This path can be viewed as minimally exposed to the attack or jamming. Another variant is when the sum of these distances is to be maximized.

Maximum data volume path. A capacity value (in bit/s) is given for each link. We also have an estimated lifetime for each link (the expected time until node movement makes the link disappear). Find a path on which the maximum expected number of bits can be sent before the path has to be updated.

Path with maximum number of round trips. A delay value is known for each link, as well as the estimated lifetime of the link. The lifetime of a path is the minimum of its link lifetimes. Find a path such that the ratio of its lifetime versus its end-to-end delay is the largest. That is, if links are assumed to be bidirectional and the delay is the same in both directions, then this path can have the maximum number of round trips during its lifetime.

Minimum energy path. For each node, two energy values are given: one that is consumed by transmitting a packet and another (typically smaller) value that is consumed when receiving a packet. Find a path such that the total enery consumed for the end-to-end delivery of a packet is minimum. The total energy takes into account every transmission and reception of the packet in the network, including receptions by neighbors that are not the intended recipients, but waste energy by overhearing the packet.

Maximum battery lifetime path. In addition to the setting of the minimum energy path, now each node has a given energy budget that is consumed by the transmission/reception of packets. Find a path that can deliver the largest number of packets before any node runs out of energy, if the considered nodes are the ones that are directly affected by the path (i.e., the nodes on the path and their neighbors).

Path with minimum error propagation. Let P be a u–v path. A set L of other links ($L \cap P = \emptyset$) is called a *backup set* for P if, for any link $e \in P$, if e is removed from the graph, then there is still a path between u and v in $(P - \{e\}) \cup L$. In other words, if any given link on path P fails, then we can still find a replacement path using extra links only from the backup set L. Task: find a path P with the smallest backup set. This path has minimum potential negative influence on the rest of the network, in the sense that it uses the smallest set of extra links to get rerouted if any link on the path fails.

Path with smallest delay sensitivity. Assume a delay value is known for each link. The *delay sensitivity* S of a u–v path P is the maximum *increase* in the end-to-end delay if a link fails on P and the path is replaced by a minimum delay path from the rest of the graph, that is, using the edges $E - \{e\}$. Formally, it is

$$S = \max_{e \in P} \min_{P'} \{\text{delay}(P') \mid P' \subseteq E - \{e\}, P' \text{ is a } u\text{–}v \text{ path}\} - \text{delay}(P)$$

Tasks: (1) Find a path with minimum delay sensitivity. (2) Restricting the search to minimum delay paths only, find a path among them with smallest delay sensitivity. (3) Find a path with minimum delay sensitivity among the paths that satisfy a given delay bound (which can be higher than the minimum delay).

Optimal multifrequency path. Modern hardware can make the nodes capable of flexibly switching between different radio frequency bands. Lower frequency links have lower speed and higher delay, but can bridge over longer distance, resulting in longer expected lifetime in a mobile ad hoc network. Switching a packet from one frequency band to another, however, requires extra processing. We also know a delay value for each link. Find a path with minimum total delay, where the total delay includes the link delays and a processing delay for frequency change along the path. The situation can be modeled by assigning weights and colors to the links (weights represent delay and colors represent the different frequency bands). In terms of this representation, we are looking for a path for which a linear combination of the path weight and the number of color changes along the path is minimum.

Combinations. The above objectives can be combined in many different ways. For example, find a minimum energy path that obeys delay and reliability constraints in a multifrequency environment.

Multiple paths. All problems can be extended to the case in which multiple paths are to be found. For example, find *disjoint* paths between the same end nodes, such that they both obey a constraint on one or more relevant parameters (such as delay, energy, reliability, delay sensitivity, expected lifetime, etc.).

These examples show that even simple static pathfinding is not always easy. For instance, the *most reliable delay-constrained path* is equivalent to another task known as the *shortest weight-constrained path* problem in which we search for a minimum weight path under the constraint that it obeys a given upper bound in another weighting. This task is known to be NP-hard [13] (and its decision version is NP-complete). Yet, a number of positive results are known for it. For example, it is solvable in pseudopolynomial time [13], that is, by an algorithm that has a polynomial running time in terms of the size of the weights, but not in terms of the number of bits that define the weights. (The size can be exponentially large compared to the number of bits.) The algorithm presented in [13] for this task runs in $O[n^5 b \log(nb)]$ time, where n is the number of nodes and b is the size of the largest weight. Thus, this algorithm becomes exponential if the weights are exponentially large, but works in polynomial time for polynomially bounded weights. There is also good news for the general case, too. Even with arbitrary weights, there is a *fully polynomial time-approximation scheme (FPTAS)* for this problem [12, 17]. An FPTAS is an algorithm that receives an error parameter ε in addition to the original input. It runs in polynomial time in terms of the input size and of $1/\varepsilon$ and produces a solution that is within $1 + \varepsilon$ times of the optimum (minimum) value.[3]

16.3 A UNIFIED REPRESENTATION OF PATH METRICS

Having seen a sample of the diversity of possible path metrics, each with its own motivation, one may naturally arrive at the question: is it possible to somehow capture all this in a unified framework? Below, we show that such a unification is indeed possible and we also present some initial results in this direction. It can be developed both for directed and undirected graphs in essentially the same way.

To define a general path metric, let us consider a family $\mathcal{F}$ of *subsets* of the edge set. This family may contain the same set with arbitrary multiplicity. We define the path metric as the number of sets in $\mathcal{F}$ that intersect with the path (counted with multiplicity). Recall that the path is meant as a set of its edges. After the formal definition, we show some examples and prove the interesting fact that essentially *any* path metric, no matter how sophisticated, can be represented in this relatively simple way.

Definition 1 ($\mathcal{F}$-measure or $\mathcal{F}$-metric). Let P be a path. Given a family $\mathcal{F}$ of sets of edges in the graph, the $\mathcal{F}$-measure (or $\mathcal{F}$-metric) of the path P is defined as the number of edge sets $H \in \mathcal{F}$ that have nonempty intersection with P. The sets are counted with multiplicity. Formally, the $\mathcal{F}$-measure of P, denoted by $\mathcal{F}(P)$, is defined as

$$\mathcal{F}(P) = \sum_{H \cap P \neq \emptyset, H \in \mathcal{F}} m(H)$$

where $m(H)$ is the multiplicity of the set H in $\mathcal{F}$. A path with minimum $\mathcal{F}$-metric is called an $\mathcal{F}$-shortest path.

Remark: There is nothing that would prevent the replacemant of integer mutliplicities by real weights. The results below carry over to this more general case as well. Nevertheless, in this introductory presentation, we stay with the combinatorial setting of integer multiplicities.

Before proceeding further, let us review a few simple examples. They may help the reader to get some feeling for the $\mathcal{F}$-measure.

Example 1: *Shortest Path.* Let $\mathcal{F}$ contain each edge (as a singleton set), with multiplicity 1. Then $\mathcal{F}(P)$ is precisely the number of edges in P, that is, the hop metric. More generally, if the multiplicity of edge e_i in $\mathcal{F}$ is some integer w_i, then $\mathcal{F}(P)$ is the weight of the path under the weighting given by the w_i values.

[3] An FPTAS is stronger than a *polynomial time-approximation scheme (PTAS)* that also runs in polynomial time in terms of the original input, with the same error bound, but may be exponential in terms of $1/\varepsilon$. Thus, error reduction can be exponentially costly for a PTAS, but not for an FPTAS.

Example 2: *Minimum Energy Path.* Let $\mathcal{F}$ contain each directed edge in a directed graph with multiplicity 2, as well as each *pair* of directed edges that have a common tail, with multiplicity 1. That is, the two-element edge set $\{e, f\}$ is in $\mathcal{F}$ if and only if $e = (u, v)$ and $f = (u, w)$ for some vertices u, v, w; $v \neq w$. It is not difficult to see that in this case $\mathcal{H}(P)$ is precisely the metric that we used in the *minimum energy path* problem, in the special case when transmission and reception both need unit energy. By changing the multiplicities in $\mathcal{F}$, one can incorporate different transmit and receive energy levels.

Example 3: *Minimum Weight k-hop Path.* For arbitrary nonnegative integers w_i, let $\mathcal{F}$ contain w_i copies of edge e_i for $i = 1, \ldots, m$, where m is the number of edges. Further, for a given integer $1 \leq k \leq m$, let $\mathcal{F}$ also contain all edge sets H that have the following properties: $|H| = m - k$ and the complement of H is a u–v path. Each such edge set H is included in $\mathcal{F}$ with multiplicity $M = 1 + \Sigma_{i=1}^{m} w_i$.

Now we can observe that the singleton sets in $\mathcal{F}$ will contribute $w(P)$ to $\mathcal{H}(P)$, where $w(P)$ is the weight of the path P with edge weights w_i. The edge sets of cardinality $m - k$ in $\mathcal{F}$ behave such that if the path has more than k edges, then it intersects every such set (since each misses only k edges). On the other hand, if the path has exactly k edges, then precisely one of the different $m - k$-element sets is disjoint from P, the one that has P as its complement. Finally, if P has less than k edges, then it again intersects all these sets. To see this, observe that a simple u–v path cannot be a proper subset of another simple u–v path (this claim will be formally proven in the proof of Theorem 1 below). Therefore, for any set $H \in \mathcal{F}$, the path P cannot be fully contained in the complement of H, so $P \cap H \neq \emptyset$ must hold. Thus, if the number of k-hop u–v paths is denoted by R, then we have

$$\mathcal{H}(P) = \begin{cases} w(P) + RM & \text{if } |P| \neq k \\ w(P) + (R-1)M & \text{if } |P| = k \end{cases}$$

Since M is chosen such that $M > w(P)$ holds for every path, therefore, the path for which $\mathcal{H}(P)$ is minimum will be precisely the one that has minimum weight $w(P)$ among the u–v paths that have exactly k edges, if such a path exists. If no such path exists, then $\mathcal{H}(P) > RM$.

(*Excercise:* Modify the above construction such that the path that minimizes $\mathcal{H}(P)$ will be a minimum weight path among those paths that have *at most k* edges, if such a path exists.)

Example 4: *Disjoint Connecting Paths with Minimum Total Hops.* Let us consider the following problem of finding *multiple* paths. Assume that two source nodes s_1 and s_2, and two terminal nodes t_1 and t_2 are given in a directed graph. We would like to find two *edge-disjoint* directed paths P_1 and P_2 such that P_1 connects s_1 to t_1 and P_2 connects s_2 to t_2 and $|P_1| + |P_2|$ is minimum. The problem is known to be *NP*-hard. In fact, just to decide if it is possible to connect the given terminals by edge-disjoint directed paths, irrespective of their lengths, is already *NP*-complete[4] [10]. On the other hand, as shown in [15], if the paths exist and the objective is to minimize the

[4] It is essential that the terminals of each path are specified. If we just look for two edge-disjoint paths to connect the *sets* $\{s_1, s_2\}$ and $\{t_1, t_2\}$, then it can be solved by network flow techniques in polynomial time.

length of the longer path, rather than the sum, then it can be approximated in polynomial time within a factor of 2. This also implies an approximation factor of at most 2 for the summed length, due to $\max\{|P_1|, |P_2|\} < |P_1| + |P_2| \leq 2\max\{|P_1|, |P_2|\}$.

Let us express the problem with the $\mathcal{F}$-measure, for the $|P_1| + |P_2| \to \min$ objective. First, if the graph does not contain the edge (t_1, s_2), then let us add this edge. Now construct $\mathcal{F}$ as follows. Add each edge, except (t_1, s_2), as a singleton set to $\mathcal{F}$. Further, let $\mathcal{H}$ be the set of s_1–t_2 paths that contain the edge (t_1, s_2). For each path $P \in \mathcal{H}$ let us add the set $\overline{P}$ (the complement of P) to $\mathcal{F}$ with multiplicity m, where m is the number of edges in the original graph. Every simple path $P \in \mathcal{H}$ is of the form $P = P_1(t_1, s_2)P_2$, that is, the concatenation of three paths: P_1 from s_1 to t_1, the edge (t_1, s_2), as well as another disjoint path P_2 from s_2 to t_2. It is not difficult to see (exercise!) that for such a P

$$\mathcal{F}(P) = |P_1| + |P_2| + m(|\mathcal{H}| - 1) \leq m|\mathcal{H}|$$

holds, whereas in case of $P \notin \mathcal{H}$ we have $\mathcal{F}(P) > m|\mathcal{H}|$. Thus, the $\mathcal{F}$-shortest path will correspond to the required pair of disjoint paths with the minimum sum of hop counts.

Having seen some examples, the reader may start wondering: what are the limits of the expressive power of the $\mathcal{F}$-measure? Can *every* path metric be expressed in this unified way by picking the appropriate family of sets for $\mathcal{F}$? Below, we show that essentially every path metric, no matter how complicated, can be represented this way. Thus, the $\mathcal{F}$-measure can serve as a canonical representation of all path metrics and, therefore, the concept of $\mathcal{F}$-shortest path is a canonical representation of any path-optimization task.

Before going into the representation, let us note that the $\mathcal{F}$-measure, by definition, can only be a nonnegative integer. As remarked after Definition 1, the model can be directly extended to the real-valued case. On the other hand, when we search for a u–v path that is optimal for some path metric $h(P)$ in a graph, the order relationship between the paths does not change if the metric $h(P)$ is replaced by $a + bh(P)$ for any constants a, b, with $b > 0$. This implies that, from the path-optimization point of view, it is enough to restrict ourselves to positive integer valued metrics. That is why we stay with integer multiplicities in this introductory exposition.

Now we can state the main representation theorem that shows that the $\mathcal{F}$-measure provides a realization of any integer-valued metric, apart from a constant translation.

Theorem 1 (Path Metric Representation Theorem). Let G be a (directed or undirected) graph with two distinguished vertices u and v that serve as endpoints of the considered paths. Let h be an arbitrary path metric that assigns a positive integer value $h(P)$ to every u–v path P in G. Then there exists a family $\mathcal{F}$ of sets of edges in G and a constant M, such that the $\mathcal{F}$-measure represents the path metric $h(P)$ for every u–v path in G, up to constant translation, that is,

$$\mathcal{F}(P) = h(P) + M$$

holds for every u–v path in G.

Proof. Let $\mathcal{P}_{u,v}$ be the set of u–v paths in G and let $R = |\mathcal{P}_{u,v}|$. Set

$$A = \sum_{P \in \mathcal{P}_{u,v}} h(P)$$

Construct the family $\mathcal{F}$ of edge sets as follows. For each $P \in \mathcal{P}_{u,v}$ include the edge set $\overline{P}$ (the complement of P) in $\mathcal{F}$ with multiplicity $A - h(P)$. Define the constant M as

$$M = (R-2)A$$

We claim that with this choice of $\mathcal{F}$ and M the relationship $\mathcal{F}(P) = h(P) + M$ holds for every u–v path in G.

First we show that a simple u–v path P be can never be a proper subset of another simple u–v path Q. Assume indirectly that $P \subseteq Q$, but $P \neq Q$. Let us traverse Q from u to v. (Note that if G is undirected, then it is not a priori necessary that the common edges of P, Q are traversed in the same direction by both paths.) In the traversal of Q, let e be the first edge in Q that is not contained in P (such an edge must exist, since P is assumed to be a proper subset of Q). Let w be the endpoint of e that is reached first in the traversal. Then $w \neq v$, since Q is not fully traversed yet when we reach w. Now we have two possibilites. (1) If $w = u$, then P must leave u on an edge $f \neq e$. But then, since Q does not visit u again, $f \notin Q$ must hold, a contradiction to $P \subseteq Q$. (2) If $w \neq u$, then let f_1 be the edge on which P arrives at w and f_2 be the edge on which it leaves. Since $e \notin P$, therefore, $f_1, f_2 \neq e$. But then at least one of f_1, f_2 cannot be in Q, since Q cannot contain three edges that are incident to w, so we again contradict $P \subseteq Q$. Thus, in either case we arrive at a contradiction, proving that an u–v path cannot be a proper subset of another one.

Now let us compute $\mathcal{F}(P)$ for an arbitrary $P \in \mathcal{P}_{u,v}$. By the construction, the sets in $\mathcal{F}$ are all of the form $\overline{Q}$ with $Q \in \mathcal{P}_{u,v}$. Observe that whenever $P \neq Q$, we have $P \cap \overline{Q} \neq \emptyset$, as we have shown that in case of $P \neq Q$ the relationship $P \subseteq Q$ cannot hold, since then P would be a proper subset of Q. Therefore, if $P \neq Q$ holds, then P must intersect the complement of Q. On the other hand, naturally, $P \cap \overline{P} = \emptyset$. Thus, P intersects all sets in $\mathcal{F}$, except the ones that correspond to its own complement. This implies, by the construction of $\mathcal{F}$ and the multiplicities,

$$\mathcal{F}(P) = \sum_{Q \in \mathcal{P}_{u,v} - \{P\}} [A - h(Q)] = (R-1)A - \sum_{Q \in \mathcal{P}_{u,v} - \{P\}} h(Q)$$

Observing that the second sum is equal to $A - h(P)$, we obtain

$$\mathcal{F}(P) = (R-1)A - [A - h(P)] = h(P) + (R-2)A = h(P) + M$$

which proves the theorem. ■

One can observe at this point that the representation provided in the proof of Theorem 1 generally uses exponentially many different sets for $\mathcal{F}$. This is the case even if a much smaller $\mathcal{F}$ would do, such as for the hop metric in Example 1. Clearly, it would be desirable to have a "small" representation whenever possible. This points to the first open question.

Open Problem 1. Characterize the path metrics that can be represented by $\mathcal{F}$-measures that have only polynomially many different sets in $\mathcal{F}$. Find a general method for constructing this "small" representation, whenever it exists.

16.4 FINDING THE $\mathcal{F}$-SHORTEST PATH

Some of the path metrics generate *NP*-hard path-optimization tasks. For example, as mentioned in Section 16.2, the *shortest weight constrained path* problem has this property. Since, as we have seen in Section 16.3, the $\mathcal{F}$-measure can represent any path metric, therefore, we cannot expect that a polynomial-time algorithm exists to find an $\mathcal{F}$-shortest path for an *arbitrary* $\mathcal{F}$. On the other hand, we show that it is possible to find an approximately $\mathcal{F}$-shortest path efficiently, if $\mathcal{F}$ is "small."

Theorem 2. Let $\mathcal{F}$ be a family of sets, given by explicit listing, in a (directed or undirected) graph with n vertices. Assume there are k sets in $\mathcal{F}$ (counted with multiplicity), each with cardinality at most r. Then there is an algorithm of complexity $O(\max\{n^2, kr\})$ that finds a path P between any two given vertices such that P is an r-approximation of the $\mathcal{F}$-shortest path, that is,

$$\mathcal{F}(P) \leq r\mathcal{F}(P_0)$$

holds, where P_0 is an $\mathcal{F}$-shortest path between the same end nodes.

Proof. Let $e_1, e_2 \ldots$ be the edges of the graph and u, v be the end vertices of the sought path. Let us define edge weights, as follows. To edge e_i assign the weight

$$w_i = \sum_{e_i \in H \in \mathcal{F}} m(H)$$

where $m(H)$ is the multiplicity of the set H in $\mathcal{F}$. In other words, w_i is the number of sets in $\mathcal{F}$, counted with multiplicity, that contain the edge e_i. Each w_i can be found and recorded in altogether $O(\max\{n^2, kr\})$ time, by simply scanning all sets in $\mathcal{F}$ and recording for each edge in how many sets it occured. Now find a minimum weight u–v path P according to this weighting. Since the path can be found in $O(n^2)$ time by Dijsktra's algorithm, therefore, the overall complexity remains $O(\max\{n^2, kr\})$.

Now we claim that the found path P has the desired property of being an r-approximation of the $\mathcal{F}$-shortest path. Let P_0 be an $\mathcal{F}$-shortest u–v path. Let us count for each edge $e_i \in P_0$ the number of sets $H \in \mathcal{F}$, counted with multiplicity, that contain e_i. Then we get precisely w_i. If we sum up the w_i weights along the path P_0, then we can get at most $r\mathcal{F}(P_0)$, since each set $H \in \mathcal{F}$ can be counted at most r times, as it contains at most r edges. On the other hand, this sum is exactly the weight $w(P_0)$ under the weighting w_i. Thus, we have $w(P_0) \leq r\mathcal{F}(P_0)$. Now observe that the path P, found as a minimum weight path under the weighting w_i, cannot have larger weight than any other u–v path. Therefore, $w(P) \leq w(P_0)$ must hold, which implies the desired relationship $\mathcal{F}(P) \leq r\mathcal{F}(P_0)$. ■

Note that in the above proof we did not really require that the sets of $\mathcal{F}$ were explicitely listed; this was only assumed for simplicity. In fact, it is enough if we have a subroutine that can count for each edge the number of sets in $\mathcal{F}$ that contain the edge.

Regarding the algorithmic issues on $\mathcal{F}$-shortest paths, two open problems are definitely worth mentioning.

Open Problem 2. Characterize the families (or find special families) $\mathcal{F}$ of edge sets for which it is possible to find an $\mathcal{F}$-shortest path in polynomial time.

Open Problem 3. In the case when no polynomial-time algorithm is available to compute an exact solution, find efficient approximations with provable bounds on the approximation error. An example in this direction is Theorem 2.

16.5 MOBILE PATHS

A proposed method for the assessment and comparison of routing protocols in mobile ad hoc networks is the *MERIT* framework [6, 7]. This method compares the route sequences found by a given routing protocol with an ideal route sequence that could be found if we knew the node movement in advance. We do not discuss the details of the entire protocol assessment framework here, as it is covered by [6, 7]. We only consider the interesting graph algorithmic issues from this framework and point out that they also fit as a special case in the $\mathcal{F}$-shortest path model.

To model mobility, the *MERIT* framework introduces the concepts of *mobile graph* and *mobile path*. A mobile graph $\mathcal{G}$ is defined as a sequence G_i, $i = 1, \dots, T$, of graphs on the same node set, where the successive graphs represent a history of the network topology changes over some time horizon T:

$$\mathcal{G} = G_1 G_2 \dots G_T$$

In a mobile graph, a *mobile path* between a source–destination s–t pair is defined as a sequence of paths

$$\mathcal{P} = P_1 P_2 \dots P_T$$

where P_i is a static path in G_i between the same source–destination s–t pair.

The weight of a mobile path $w(\mathcal{P})$ includes two basic components:

1. The weights $w_i(P_i)$ of the individual static paths, according to arbitrary individual weightings in the graphs G_i
2. Some transition cost $c_{\text{trans}}(P_i, P_{i+1})$ incurred by the protocol whenever there is a change in the path sequence

Thus, the weight (cost) of a mobile path $\mathcal{P}$ is defined as

$$w(\mathcal{P}) = \sum_{i=1}^{T} w_i(P_i) + \sum_{i=1}^{T-1} c_{\text{trans}}(P_i, P_{i+1}) \tag{16.1}$$

The *transition cost* between paths is a cost function associated with having to update from one path to another in the routing protocol. In general, it is the overhead associated with updating the routing state to reflect the change in the path.

Now we can define the *shortest mobile path problem* as follows.

Shortest Mobile Path (SMP) Problem: Given a mobile graph $\mathcal{G} = G_1 \ldots G_T$ and a specified source–destination pair s, t, find a mobile path $\mathcal{P} = P_1 \ldots P_T$ from s to t, such that the weight

$$w(\mathcal{P}) = \sum_{i=1}^{T} w_i(P_i) + \sum_{i=1}^{T-1} c_{\text{trans}}(P_i, P_{i+1})$$

of the mobile path is minimum.

Two fundamental results were first proven about mobile paths in [6]. The first is a positive result, as follows. Let the transition cost function be simply an indicator of the path change, i.e., given as

$$c_{\text{trans}}(P_i, P_{i+1}) = \begin{cases} 0 & \text{if } P_i = P_{i+1} \\ c_0 & \text{if } P_i \neq P_{i+1} \end{cases} \tag{16.2}$$

where c_0 is a positive constant. That is, no transition cost is incurred if there is no change in the path in successive graphs, otherwise a constant cost is incurred if the path has changed. In this case, the shortest mobile path can be found by a dynamic programming algorithm that runs in polynomial time. This makes it possible to compute the ideal route sequence that is used in the *MERIT* framework as a basis of comparison.

The second result is negative: if the transition cost can be arbitrary (but still efficiently computable), then the problem becomes *NP*-hard. Specifically, it is already *NP*-hard for the following transition cost function:

$$c_{\text{trans}}(P_i, P_{i+1}) = M(|P_i \cap P_{i+1}| + |w_i(P_i) - w_{i+1}(P_{i+1})|)$$

where M is a constant.

Now we show that mobile paths also fit in the unified path metric framework presented in Section 16.3, thus, the shortest mobile path problem is a special case of the $\mathcal{F}$-shortest path problem. For simplicity, let us assume that the weights and transition costs are positive integers (this is not an essential restriction).

Assume that we are given an instance of the mobile path problem as described above. Let s and t be the source and terminal nodes of the sought mobile path, respectively. (These nodes are present in each G_i.) Let us construct a new graph by taking each graph G_i in the sequence and merge the end node t in G_i with the start node s in G_{i+1}. Let us call the resulting new graph G. Let the start node in the first graph G_1 be named u and the end node in the last graph G_T be named v. Now it is clear that every path sequence (mobile path) is in 1–1 correspondence with an u–v path in the amalgamated graph G. (The path in G is simply the concatenation of the path sequence in the mobile path.)

According to the above construction, we can uniquely represent each possible mobile path by a u–v path in G. Let us also keep the original edge weights. It would not be sufficient, however, if one simply looked for a minimum-weight u–v path in the amalgamated graph G, since that would ignore the transition costs. We can define, however, a path metric by Equation (16.1) for every u–v path in G. This metric includes both the static weight and the transition costs. Clearly, P_i means here the part of the path that falls in G_i. Now we can use the fact that, by Theorem 1, *every* path metric can be represented by an $\mathcal{F}$-measure, up to constant translation. Thus, we have that the shortest mobile path problem is equivalent to an $\mathcal{F}$-shortest path problem. In this sense, the shortest mobile path also becomes a special case of the general framework presented in Sections 16.3 and 16.4.

What is the advantage of translating the shortest mobile path problem into an $\mathcal{F}$-shortest path task? The gain is that any result/algorithm for the $\mathcal{F}$-measure will automatically apply to mobile paths. For example, if we represent the mobile path cost [Equation (16.1)] by an $\mathcal{F}$-measure (made possible by Theorem 1), then we can approximate this $\mathcal{F}$ by a smaller family that allows a good approximation algorithm by Theorem 2. This yields the research question:

Open Problem 4. Find a way to approximate the mobile path measure [Equation (16.1)] with a "small" $\mathcal{F}$-measure.

One can also use the special structure of the SMP problem in a different way. Let us assume that instead of allowing any s–t path in G_i, we restrict ourselves in each G_i to a preselected set $\mathcal{P}_i$ of paths. It is not unusual in practical situations that only a restricted set of paths is considered. Note that this still allows exponentially many mobile paths, since if $|\mathcal{P}_i| = R_i$, then we can still have $R_1 R_2 \cdot \ldots \cdot R_T$ different mobile paths. Now we can define a restricted version of the SMP problem as follows.

Restricted Shortest Mobile Path (R-SMP) Problem: The input is a mobile graph $\mathcal{G} = G_1 \ldots G_T$ and a specified source–destination pair s–t, as well as a set of s–t paths $\mathcal{P}_i$ for each G_i. Find a mobile path $\mathcal{P} = P_1 \ldots P_T$ from s to t, such that the restriction $P_i \in \mathcal{P}_i$ holds for each i and the weight

$$w(\mathcal{P}) = \sum_{i=1}^{T} w_i(P_i) + \sum_{i=1}^{T-1} c_{\text{trans}}(P_i, P_{i+1})$$

of the restricted mobile path is minimum.

Now we show that the above-defined restriction is quite useful: the R-SMP problem behaves in a much more "friendly" way than the original SMP. Specifically, the restricted shortest mobile path can be found in polynomial time for *any* transition cost function, given that the path sets $\mathcal{P}_i$ are of polynomially bounded size. The details are shown in the following theorem.

Theorem 3. Let $|\mathcal{P}_i| = R_i$ in the R-SMP problem and set $R = R_1 + \ldots + R_T$. Then, for any transition cost function the shortest restricted mobile path can be found in $O(R^2)$ time, assuming that the transition cost function is computed by a subroutine in unit time. In particular, if R is polynomially bounded in terms of the size of the mobile graph, then the solution is obtainable in polynomial time.

Proof. Let us construct an auxiliary directed graph G' as follows. Let us denote the paths in $\mathcal{P}_i$ by $P_{i,1}, P_{i,2}, \ldots, P_{i,R_i}$, where $R_i = |\mathcal{P}_i|$. For each $P_{i,j}$ take two nodes $a_{i,j}$, $b_{i,j}$, draw an edge form $a_{i,j}$ to $b_{i,j}$ and assign the weight $w_i(P_{i,j})$ to this edge. Further, connect each node $b_{i,j}$ to every $a_{i+1,k}$ by an edge, directed toward $a_{i+1,k}$, and assign the weight $c_{\text{trans}}(P_{i,j}, P_{i+1,k})$ to this edge. Finally, add two extra nodes a, b and draw a directed edge from a to each $a_{1,j}$, as well a directed edge from each $b_{T,j}$ to b. All edges that are adjacent to a or b are assigned 0 weight. The whole construction can be done in $O(R^2)$ time, assuming that each transition cost computation is counted as one step.

It is easy to see that every *a–b* path in G' is in a natural 1–1 correspondence with a restricted mobile path in $\mathcal{G}$. (The static component paths of mobile path are marked by the indices of nodes that the *a–b* path in G' traverses.) Moreover, the weight of any *a–b* path in G' is precisely the weight of the corresponding mobile path, including transition costs. Thus, to solve the R-SMP problem optimally, one only needs to find a minimum weight *a–b* path in G'. Since G' has $R + 2$ vertices, this can be done in $O(R^2)$ time by Dijkstra's algorithm. ■

As seen above, once the preselected path sets $\mathcal{P}_i$ are given, the R-SMP problem can be solved in a rather straightforward way. A key question, however, remains open:

Open Problem 5. Find a method to choose the preselected path sets, such that the R-SMP solution provides a good approximation to the SMP.

16.6 THE INVERSE SHORTEST PATH PROBLEM

It is not an unusual case in ad hoc routing protocols that the declared path metric of a protocol (which is most often the simple hop metric) is not the one that really reflects the actual choices. The reason is that there is interaction among the various layers in the protocol stack and the influence of protocols in other layers (for example, the effect of TCP) makes certain routes less preferable or even unavailable. The routing protocol can take this into account via auxiliary mechanisms, for example, maintaining various preferences, timeout parameters, and so on, rather than explicitely incorporating it into a numerically defined path metric. As a result, the *actual* route choices can reflect a metric that is not defined explicitely. Rather, it is the result of the interaction between the declared path metric *plus* a number of auxiliary mechanisms that may depend on a large number of various factors, influenced by the behavior of other layers.

This *impliciteness* of the actual path metric complicates the analysis and assessment of routing protocols. It is not easy to decide whether the protocol choses the "best" (static) path if we do not even have an explicitely defined path metric. Below, we outline an approach that may be of help to handle this issue.

In the *Inverse Shortest Path (ISP) Problem,* we reverse the usual setting of path finding. In the usual setting, a metric is given and the task is to find shortest paths between various end nodes, according to the given metric. In the ISP problem, the task is just the opposite: we are given the chosen paths and we would like to find the metric that makes them shortest paths among their terminal nodes. Thus, if the ISP is solved, then a metric is constructed that explicitely represents the *actual* path choices of the protocol as shortest paths, incorporating the potential effect of auxiliary mechanisms.

It is worth noting at this point that if *any* metric can be selected to represent the observed path choices of a protocol as shortest paths, then the problem is essentially trivial, since we can simply assign a small value to the given paths and a large value to every other path. This wold obviously make the given paths the shortest. On the other hand, this "brute force" aproach is not a scalable solution, since the metric has to "remember" each chosen path. What we would like to achieve is to represent the metric in a way that does not get more complicated with the growing number of paths.

A natural choice is to restrict ourselves to a metric that is generated by positive *link weights*. It has a number of advantages: it is scalable, as it does not depend on the (potentially exponential) number of paths, and it is algorithmically well tractable to find the shortest path under this metric.

A potential drawback is, however, that not all path systems can arise as shortest paths under some positive link weighting. (*Exercise:* construct such an example, that is, a set of paths in a graph for which there is no positive edge weigting that makes all the given paths shortest between their endpoints.) Therefore, we extend the task such that whenever no weighting exists to *precisely* capture the path system, then we look for a *best approximation,* as defined below.

The following formulation is based on a more general framework presented in [8], specialized here to shortest paths. For notational convenience, the edge weights are collected into a weight vector w. The weight of any given path P is denoted by $w(P)$ and the weight of a shortest (= minimum weight) path between the endpoints of P is denoted by $s_w(P)$. That is, $s_w(P) = \min_Q w(Q)$, where the minimum is taken over all paths Q that have the same endpoints as P.

Inverse Shortest Path (ISP) Problem

Input: Given paths $P_1, \dots, P_r$ among arbitrary (possibly different) end vertices in a graph G.

Find: A weight vector $w \geq 1$ that minimizes the maximum absolute error

$$\Delta = \max_i |w(P_i) - s_w(P_i)|$$

Note that the constraint $w \geq 1$ is needed for normalization purposes, otherwise one could trivially make the error arbitrarily small by assigning sufficiently small positive weights to everything. We also note that the absolute value can be omitted from the definition of Δ, as $w(P_i) \geq s_w(P_i)$ must always hold by definition.

Now we show that the ISP problem can be solved in polynomial time. The solution provides the required best approximation. Moreover, if an *exact* solution exists, that is, there is a positive weighting that makes all the given paths shortest among their respective endpoints, then the the algorithm finds this, since then the achievable best error is $\Delta = 0$.

Theorem 4. The ISP problem can be solved in polynomial time, that is, an optimal weighting w can be found in polynomial time that minimizes the maximum absolute error under the conditions of the ISP Problem.

Proof. For easy notation, let us denote the edges by the numbers $1, 2, \dots, m$. For each path P_k, given as input, let $\mathcal{P}_k$ be the set of all paths between the endpoints of P_k. Now consider the following linear program.

$$y \rightarrow \max! \tag{16.3}$$

$$y \leq 0 \tag{16.4}$$

$$w_i \geq 1 \qquad (i = 1, \dots, m) \tag{16.5}$$

$$\sum_{j \in P} w_j - \sum_{j \in P_k} w_j \geq y \qquad (\forall P \in \mathcal{P}_k - \{P_k\}, k = 1, \dots, m) \tag{16.6}$$

Note that as $\Sigma_{j \in P} w_j = w(P)$, we have just used the sums to emphasize that the above formulation is indeed a *linear* program. Below, we are going to show that the optimal solution of this linear program provides precisely the required weightings.

First, observe that this system of linear inequalities can contain exponentially many inequalities (since the sets $\mathcal{P}_k$ can be exponentially large) and, additionally, they are not even explicitly listed. Nevertheless, we can still solve it in polynomial time if we apply those linear programming algorithms that do not need an explicit list of the inequalities; they can work with a so-called *separation oracle*. A separation oracle is a subroutine that for any given variable vector can check if it satisfies all inequalities, or, if not, it return a violated inequality. A well-known method of this type is the Ellipsoid Algorithm, the historically first polynomial-time solution for linear programming (for an overview see, e.g., [20]). It is known that the Ellipsoid Algorithm with separation oracle runs in time that is polynomially bounded in terms of the number of variables, the number of bits that describe any given inequality, and the running time of the separation oracle. In our case, the separation oracle means that for any given setting of the weights w_i and the variable y, we should be able to check in polynomial time whether the above constraints are all satisfied, or, if not, we have to find a violated inequality. If we can do this, then the Ellipsoid Algorithm finds the solution in polynomial time, since our inequalities have all 0, 1, –1 coefficients and constants.

Now we construct the needed polynomial-time separation oracle as follows. Given a vector (w, y), first we directly check if Equations (16.4) and (16.5) are satisfied. If not, we have directly found a violated inequality. If they are satisfied, then we proceed as follows. For each index k run a shortest path algorithm (e.g., Dijkstra's), with weighting w, to find a shortest path between the endpoints of P_k, where P_k is the kth given input path. Assume that for a given k and w, the returned minimum weight path is P. Let us compare $w(P_k)$ and $w(P)$. If $w(P) \geq y + w(P_k)$, then all inequalities with this k must be satisfied in Equation (16.6), since P is a minimum-weight path under this weighting, so $w(P') \geq w(P)$ holds for any other path P' between the same endpoints. On the other hand, if $w(P) < y + w(P_k)$, then this provides a violated inequality, namely

$$\sum_{j \in P} w_j - \sum_{j \in P_k} w_j \geq y$$

is violated.

Having constructed the polynomial-time separation oracle, we can find an optimum solution (w_0, y_0) to the linear program by the Ellipsoid Algorithm in polynomial time. It follows from the construction that $|y_0|$ will be the smallest possible value for which $w_0(P_k)$ differs form $s_w(P_k)$ $(\forall k)$ at most by $|y_0|$, which proves the theorem. ■

The algorithm described in the above proof solves the ISP problem in polynomial time, but it is still not too practical, as it involves linear programming in the special form in which instead of explicitely listed inequalities one has to use implicitely defined constraints, in terms of a separation oracle (although it runs in polynomial time, most linear programming software packages do not support this mode of operation; they typically need explicite constraints). This yields the following open question.

Open Problem 6. Find a method to solve the inverse shortest path problem in a purely combinatorial way, that is, as a pure graph algorithm that avoids linear programming.

Another related question is the following:

Open Problem 7. Provide (a preferably simple) characterization of the path systems that can be exactly represented as shortest paths under some positive weighting. Is there a pure graph theoretic characterization that is not based on linear programming?

16.7 ROUTE SYSTEMS

Viewing a route in isolation does not tell too much about its contribution to network performance, since the network serves many sessions in parallel. It is a more difficult question, however, how to find an entire *route system* among various nodes that is "good" in any well-defined sense.

A typical problem in this context is to find routes that connect given node pairs, such that available capacity (bandwidth) bounds are obeyed on the links. The task can be transformed into a purely graph theoretic problem by replacing any edge i of capacity C_i by C_i parallel edges (C_i is assumed to be an integer) and then the task becomes to search for edge disjoint paths between given endpoints in the transformed graph.

This *disjoint connecting paths* problem, that is, to decide whether or not a given set $(s_1, t_1), \dots, (s_k, t_k)$ of terminator pairs can be connected via edge-disjoint paths $P_1, \dots, P_k$ such that P_i connects s_i with t_i for each i, is one of the classical *NP*-complete problems [11] that appeared in the sources of the *NP*-completess theory among the original problems of Karp [14]. It remains *NP*-complete both for the directed and undirected, as well as the edge disjoint and vertex disjoint paths versions. The corresponding natural optimization problem, when we are looking for the maximum number of terminator pairs that can be connected by disjoint paths is *NP*-hard.

The restriction in *disjoint connecting paths* that we are looking for paths that connect each source node with a dedicated destination is essential. If this is relaxed and we are satisfied with edge-disjoint paths that connect each source s_i with *some* of destinations t_j but not necessarily with t_i, then the problem becomes solvable with classical network flow techniques. Thus, the prescribed matching of sources and destinations causes a dramatic change in the problem complexity.[5]

Considerable research has been done on various decision and optimization versions of the *disjoint connecting paths* problem in the discrete mathematical and theoretical computer-science community, leading to a good number of deep theorems and approximations with provable properties (see, e.g., [2, 3] and further references therein). Unfortunately, most of the theoretical results are rather complex (to implement or even to understand), yielding a situation in which practical applications hardly use anything other than the simplest greedy heuristic, at least when fast solution is needed.

An additonal factor in ad hoc networks is that one cannot reasonably assume that each node has full information about all the route requests in the entire network. Moreover, the requests are not simultaneous. A model that attempts to capture this issue in an exact way is *on-line route search*. In on-line route search algorithms, it is assumed that the route requests are ordered in time and they come one by one. We have to find a route for each before receiving the next request. That is, we can only use past information; future requests

[5] Interestingly, according to an old result, it becomes *NP*-complete if we require that just *one* of the sources is connected to a dedicated destination; the rest is relaxed as above [5].

are not known in advance, which is a natural assumption. Additionally, the information that is available about the past may be restricted when chosing the next route.

The performance of such an on-line algorithm is measured by the so-called *competitive ratio*. Let us consider, for example, the objective of selecting the routes such that the maximum link load (congestion) is minimized. In this context, an on-line algorithm is called *$f(n)$-competitive* for some function $f(n)$ of the network size n if it guarantees a solution in which the maximum link load is asymptotically at most $f(n)$ times higher than the value that could be achieved by *any* algorithm, even by an optimal off-line one that knows the whole request sequence in advance. Thus, the competitive ratio essentially tells us how much we have to pay in performance degradation for not knowing the future in advance.

The best achievable competitive ratio in on-line route search is known to be $f(n) = O(\log n)$, where n is the number of nodes [1, 18]. Interestingly, it can be realized by a surprisingly simple algorithm, as follows.

The idea is that the weight of any given link is an exponential function of the current load of the link. More precisely, let C_e be the capacity of link e and let V_i be the requested capacity for the ith route request R_i. Let $r_e(i-1)$ denote the *relative load* that has been accumulated on link e while routing the first $i-1$ requests, that is, the summed capacity of the paths that have been routed through e before R_i, divided by the capacity of the link:

$$r_e(i-1) = \frac{\sum_{j=1}^{i-1} V_j}{C_e}$$

After having routed the first $i-1$ requests, the weight w_e of link e is updated according to the following formula:

$$w_e = \mu^{re(i-1)}(\mu^{Vi/Ce} - 1) \tag{16.7}$$

where $\mu > 1$ is a constant. Thus, the algorithm can be simply described as follows.

Algorithm *On-line Route Search*

```
Step 1  Initialization: set all link weights to 1; $i := 1$.
Step 2  Find a minimum-weight path between the given terminators of $R_i$, according
          to the current weights.
Step 3  Recompute the weights according to Equation (16.7).
Step 4  If all requests are routed then stop, else $i := i + 1$, go to Step 2.
```

The above algorithm is proven to be $O(\log n)$-competitive for a network of n nodes [18]. Recall that it means the algorithm guarantees a routing in which the maximum link load is (asymptotically) at most $O(\log n)$ times higher than the value which could be achieved by *any* algorithm, even by an optimal off-line one that knows the whole request sequence in advance.

It is a remarkable feature of the above algorithm that it uses only the aggregated relative load values of the links, but it does not require detailed information on the past route requests. On the other hand, the algorithm allows that the link capacity may be exceeded by the load, so the capacity is not viewed as a hard constraint. (We may assume that connections can slow down their speed in such a case.) This, of course, can be easily avoided if we modify the weights by setting the weight of saturated links to infinite. Unfortunately, however, this modification destroys the proof of the $O(\log n)$ competitive ratio.

It is interesting to note that even if we know the whole request sequence in advance, no polynomial time algorithm is known (to the author's knowledge) that would achieve a better than $O(\log n)$ ratio to the optimum.

Open Problem 8. Is it possible to construct a similarly simple and efficient algorithm that obeys link capacity constraints as hard limits, yet still has the same competitive ratio?

Open Problem 9. Although Algorithm On-line Route Search does not use the details of past connections to lay out the next route, it still needs to know the relative load on each link, which still requires global information about the network. Is it possible to achieve similar preformance using less global information?

16.8 CONCLUSION

It is the author's hope that this chapter will convince the reader that ad hoc networking (and, of course, networking in general) can present a number of interesting, novel challenges for algorithm development and analysis. We only had space to address some selected elements of a single area (routing). This was chosen because it perhaps offers the richest set of challenges that are directly related to graph algorithms, and, therefore, it is the easiest to follow and visualize. Nevertheless, we should not forget that the *master problem,* defying even any precise definition, is this: how do the various metrics, objectives, algorithms, and other particular elements influence the overall network performance?

REFERENCES

1. J. Aspnes, Y. Azar, A. Fiat, S. Plotkin, and O. Waarts, "On-line Machine Scheduling with Applications to Load Balancing and Virtual Circuit Routing," in *ACM Symposium on Theory of Computing (STOC'93),* pp. 623–631, 1993.
2. G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, and M. Protasi, *Complexity and Approximation,* Springer Verlag, New York, 1999.
3. A. Baveja and A. Srinivasan, "Approximation Algorithms for Disjoint Paths and Related Routing and Packing Problems," *Mathematics of Operations Research, 25,* 2, pp. 255–280, 2000.
4. R. Bellman, "On a Routing Problem," *Quarterly of Applied Math., 16,* 1, 87–90, 1958.
5. A. Faragó, "Algorithmic Problems in Graph Theory," *Conference of Program Designers,* pp. 61–66, Eötrös University, Budapest, Hungary. 1985.
6. A. Faragó and V. R. Syrotiuk, "MERIT: A Unified Framework for Routing Protocol Assessment in Mobile Ad Hoc Networks," in *Proceedings of 7th Annual International Conference on Mobile Computing and Networking* (Mobicom'2001), Rome, Italy, July 2001.
7. A. Faragó and V. R. Syrotiuk, "MERIT: A Scalable Approach for Protocol Assessment," Invited paper, *Mobile Networks and Applications (MONET),* Special Issue on Mobile Ad Hoc Networks, in press.
8. A. Faragó, Á. Szentesi, and B. Szviatovszki, "Inverse Optimization in High Speed Networks," *Discrete Applied Mathematics,* Special Issue on Combinatorial and Algorithmic Aspects of Telecommunications, in press.

9. E. W. Dijkstra, “A Note on Two Problems in Connection with Graphs,” *Numerische Mathematik,* 1, 269–271, 1959.
10. S. Fortune, J. Hopcroft, and J. Wyllie, “The Directed Subgraph Homeomorphism Problem,” *Theoretical Computer Science,* 10, 2, 111–121, 1989.
11. M. R. Garey and D. S. Johnson, *Computers and Intractability,* W. H. Freeman and Co., San Francisco, 1983.
12. R. Hassin, “Approximation Schemes for the Restricted Shortest Path Problem,” *Mathematics of Operations Research, 17,* 36–42, 1992.
13. J. M. Jaffe, “Algorithms for Finding Paths with Multiple Constraints,” *Networks,* 14, 95–116, 1984.
14. R. M. Karp, “Reducibility Among Combinatorial Problems,” in R. E. Miller and J. W. Thatcher (Eds.), *Complexity of Computer Computations,* Plenum Press, New York, 1972.
15. C. Li, S. T. McCormick, and D. Simchi-Levi, “The Complexity of Finding Two Disjoint Paths with Min-max Objective Function,” *Disc. Appl. Math.* 26, 105–115, 1990.
16. C. E. Perkins (Ed.), *Ad Hoc Networks,* Addison-Wesley, Reading, MA, 2001.
17. C. A. Phillips, “The Network Inhibition Problem,” in *Proceedings of 25th Annual ACM Symposium on Theory of Computation (STOC’93),* pp. 776–785, 1993.
18. S. Plotkin, “Competitive Routing of Virtual Circuits in ATM Networks,” *IEEE Journal Selected Areas in Communications,* 13, 1128–1136, 1996.
19. E. M. Royer and C.-K. Toh, “A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks,” *IEEE Personal Communications,* 46–55, April 1999.
20. A. Schrijver, *Theory of Linear and Integer Programming,* Wiley, New York, 1990.
21. C.-K. Toh, *Ad Hoc Mobile Wireless Networks: Protocols and Systems,* Prentice Hall, Upper Saddle River, NJ, 2001.

INDEX

Mobile Ad Hoc Networking. Edited by Basagni, Conti, Giordano, and Stojmenovic.
ISBN 0-471-37313-3 © 2004 Institute of Electrical and Electronics Engineers, Inc.

ABOUT THE EDITORS

Stefano Basagni received a Ph.D. in Computer Science from the University of Milan, Italy in 1998 and a Ph.D. in Electrical Engineering from the University of Texas, Dallas in 2001. He received his Bachelor of Science in Computer Science from the University of Pisa, Italy, in 1991. Since January 2002, he has been assistant professor of Computer Engineering in the Electrical and Computer Engineering Department of Northeastern University. Dr. Basagni's interests include research and implementation aspects of mobile networks and wireless communication systems, with an emphasis on the design and implementation of protocols for personal area networks, ad hoc networking, and their enabling technologies (Bluetooth, IEEE 802.11, etc.). In these fields, Dr. Basagni has coauthored more than 36 papers and published in peer-reviewed international journals and conference proceedings. Dr. Basagni served as a guest editor of the special issue of the *Journal on Special Topics in Mobile Networking and Applications (MONET) on Multipoint Communication in Wireless Mobile Networks,* as well as co-guest editor (with Dr. S. J. Lee) of the special issue on ad hoc networking for the Wiley-Interscience journal *Wireless Communications & Mobile Computing (WCMC).* He has served on Technical Program and Organizing committees for leading conferences, such as the ACM/SIGMOBILE MobiCom and ACM/SIGMOBILE MobiHoc. He has also served as session chair and organizer for IEEE, ACM, and IASTED conferences. Dr. Basagni has been a reviewer for several international journals, including the SIAM *Journal on Computing*, ACM/IEEE *Transactions on Networking, IEEE Journal on Selected Areas in Communications, Information Processing Letters*, *IEEE Transactions on Vehicular Technology,* ACM/Kluwer *Wireless Networks*, and ACM/Kluwer *MONET and Computer Networks.* Dr. Basagni is a member of the ACM, the IEEE, ACM SIGMOBILE and the IEEE Communication and Computer societies.

Marco Conti received the Laurea degree in Computer Science from the University of Pisa, Italy, in 1987. In 1987, he joined the Italian National Research Council (CNR). He is currently a senior researcher at CNR-IIT. His research interests include Internet architec-

Mobile Ad Hoc Networking. Edited by Basagni, Conti, Giordano, and Stojmenovic.
ISBN 0-471-37313-3 © 2004 Institute of Electrical and Electronics Engineers, Inc.

ture and protocols, wireless networks, ad hoc networking, mobile computing, and QoS in packet switching networks. He co-authored the book, *Metropolitan Area Networks*. He has published in journals and conference proceedings more than 120 research papers related to design, modeling, and performance evaluation of computer-network architectures and protocols. He served as TPC chair of the IFIP-TC6 Conferences "Networking2002," and "PWC2003," and as TPC co-chair of *ACM WoWMoM 2002,* the First IFIP-TC6 Conference on *Wireless On-demand Network Systems* (WONS 2004), and the 2nd Workshop on *Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks* (WiOpt '04). He is serving as program co-chair of the First International Workshop on Broadband Wireless Multimedia (BroadWIM 2004), vice program co-chair for the 1st IEEE Conference on Mobile Ad-Hoc and Sensor Systems (MASS-2004), and Workshops Co-chair for IEEE PerCom 2005. He is on the editorial board of *Ad Hoc Networks* journal and ACM *Mobile Computing and Communications Review.* He served as guest editor for the *Cluster Computing Journal* special issue on mobile ad hoc networking, *IEEE Transactions on Computers* special issue on "Quality of Service issues in Internet Web Services, *ACM/Kluwer Mobile Networks & Applications* Journal special issue on mobile ad hoc networks, and the "Networking2002" journal special issues on: *Performance Evaluation, Cluster Computing* and *ACM/Kluwer Wireless Networks* Journals. He is member of IFIP WGs 6.2, 6.3, and 6.8.

Silvia Giordano has a Ph.D. from the Ecole Polytechnique Federale de Lausanne (EPFL), Switzerland. She is currently working as senior researcher at the University of Applied Science—SUPSI in Ticino, Switzerland. She is teaching several courses in the areas of wireless and mobile networking, quality of services, and networks applications. Previously, she was on the faculty of the EPFL and the University of Pisa. Since October 2001, she has also been an associate researcher at CNR, Pisa. She has published extensively in journals, magazines, and conferences in the areas of quality of services, traffic control, wireless and mobile ad hoc networks. She has participated in several European ACTS/IST projects and European Science Foundation (ESF) activities. She was invited by the ESF to participate at the ERCIM-PESC meeting held at CWI, Amsterdam, May 2002, for developing a joint vision for the future of e-Science. Since 1999, she has served as technical editor of *IEEE Communications Magazine,* and is currently the series co-editor of the new series on adhoc and sensor networks of the *IEEE Communication Magazine.* She has coedited several special issues of *IEEE Communications Magazine* and *Baltzer MONET* and *Cluster Computing* on mobile ad hoc networking and QoS networking. She will be general chair of the 2005 edition of IFIP conference WONS (Wireless On-demand Network Systems) and has served on the executive committee and TPC of several international conferences. She also serves as reviewer on transactions and journals. She is a member of IEEE Computer Society and IFIP WG 6.8. Her current research interests include QoS and traffic control and wireless and mobile ad hoc networks.

Ivan Stojmenovic received Bachelor of Science and Master of Science degrees in 1979 and 1983, respectively, from the University of Novi Sad, Yugoslavia, and a Ph.D. in Mathematics in 1985 from the University of Zagreb. He earned a third degree prize at International Mathematics Olympiad for high school students in 1976. In 1980, he joined the Institute of Mathematics, University of Novi Sad and in 1988, joined the faculty in the Computer Science Department at the University of Ottawa, Canada, where he currently holds the position of a full professor in SITE. Since June 2000, he is frequently in Mexico

City as a researcher for DISCA, IIMAS, Universidad Nacional Autonoma de Mexico. He has published four books and more than 150 papers in journals and conferences. His research interests are wireless networks, parallel computing, multiple-valued logic, evolutionary computing, neural networks, combinatorial algorithms, computational geometry, and graph theory. He is currently a managing editor of *Multiple-Valued Logic*, an international journal, and an editor of the following journals: *Parallel Processing Letters, IASTED International Journal of Parallel and Distributed Systems,* and *Tangenta*. He has edited the *Handbook of Wireless Networks and Mobile Computing* (Wiley, 2002), organized two workshops on wireless networks and mobile computing at IEEE HICSS conference, and guest-edited special issues for the journals *Telecommunication Systems* and *Wireless Communications and Mobile Computing.*